# Transition between graphical and algebraic representatives of Bayesian network models (extended version)

Milan Studený[1] Jiří Vomlel[2,1]

[1]Inst. of Inf. Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4,
182 08 Prague 8, Czech Republic

[2]Laboratory for Intelligent Systems
University of Economics
Ekonomická 957,
148 01 Prague 4, Czech Republic

**Abstract**

In learning Bayesian networks one meets the problem of non-unique graphical description of the respective statistical model. One way to avoid this problem is to use special chain graphs, named *essential graphs*. An alternative algebraic approach uses certain integer-valued vectors, named *standard imsets*, instead. In this paper we present algorithms which make it possible to transform graphical representatives into algebraic ones and conversely. A direct formula is a basis for translation of any chain graph equivalent to a Bayesian network into a standard imset. An inverse algorithm, which gives the essential graph, has two stages. The middle result of this procedure is a certain sequence of sets of variables, which can be turned into a hierarchical junction tree. We present both the mathematical theory and the algorithms, which we implemented in the R language.

## 1 Introduction

General motivation for this paper is learning Bayesian networks. The goal of a learning procedure is, given data, to determine the most suitable statistical model (= a class of distributions) among those which are attributed to Bayesian networks (= acyclic directed graphs). One usually tries to maximize a certain quality criterion which ascribes the same value to equivalent graphs, that is, graphs defining the same statistical model. Such a criterion is often called a *score-equivalent criterion*. A natural problem one has to deal with is how to represent the considered statistical models in the memory of a computer. Although some authors do not insist on their unique representation and use any Bayesian network for this purpose (Kočka and Castelo, 2001), a viable approach is to use some uniquely determined representatives.

The most popular graphical representative of a Bayesian network model is the *essential graph* of the respective equivalence class of acyclic directed graphs. It is a chain graph describing shared features of acyclic directed graphs in the equivalence class. The term "essential" graph was proposed by Andersson, Madigan and Perlman (1997a) who also gave basic graphical characterization of graphs that are essential graphs. These graphs got different names in the literature: completed patterns (Verma and Pearl, 1991), maximally oriented graphs for patterns (Meek, 1995) and completed pdags (Chickering, 2002). It has been shown recently (Studený, 2004a; Roverato, 2005) that the essential graph of an equivalence class of acyclic directed graphs can be characterized as the largest chain graph within the class of equivalent chain graphs without flags (i.e., without certain special induced subgraphs).

Another possible representative of a Bayesian network statistical model is a certain integer-valued vector, named a *standard imset*. This algebraic representative plays an important role in an algebraic method for describing probabilistic conditional independence structures presented in (Studený, 2001; Studený, 2005). The advantage of this approach is that usual quality criteria used in practice appear to be shifted linear functions of these vectors – for details see § 8.4 of (Studený, 2004c; Studený, 2005).

The topic of this paper is the transition between these two representatives of a Bayesian network model. One of the motives for the effort to establish relation between them is as follows. The concept of inclusion neighbourhood (Kočka and Castelo, 2001; Auvrey and Wehenkel, 2002; Studený, 2004b) plays an important role in the method of local search (Bouckaert, 1995; Chickering, 2002; Castelo, 2002), which is used to maximize a quality criterion. The above mentioned algebraic approach is particularly suitable in this situation. This is because moves from a given state (= Bayesian network model) to its inclusion neighbours and the respective changes in the value of a quality criterion are characterized by certain elementary vectors – for details see § 8.4 of (Studený, 2004c; Studený, 2005). At present, no characterization of the collection of all inclusion neighbours of a given Bayesian network model in terms of the standard imset is available. On the other hand, its characterization in terms of the essential graph is available (Studený, 2004b).

Transition from a graphical representative to an algebraic one is relatively simple. We give a formula for the standard imset on basis of any chain graph which defines the respective statistical model.

The inverse transition is more complex. First, we introduce a series of characteristics of chain graphs equivalent to Bayesian networks. They allow one to design a reconstruction algorithm for the essential graph. It has two parts. In the first stage, a certain sequence of sets of variables is obtained – these sets appear to be cliques of closure graphs for components of the essential graph. The second stage is the proper reconstruction algorithm for the essential graph. Moreover, the sequence of sets can be utilized to constitute a certain hierarchical tree (Puch et al., 2004), which can also be viewed as a (non-unique) representative of the respective Bayesian network statistical model.

Basic concepts and facts are recalled in Section 2 where we define the standard imset for an acyclic directed graph. A general formula for the standard imset is given in Section 3; it is applicable to any chain graph equivalent to a Bayesian network. Then, in Section 4, the above mentioned characteristics of chain graphs are introduced. In Section 5 we formulate lemmas on which the reconstruction algorithm is based. The algorithm is given in Section 6. In Section 7 we discuss the relation to hierarchical junction trees. We implemented all mentioned algorithms in the R language (R Development Core Team, 2004). In Conclusions we discuss future perspectives.

## 2 Basic concepts

In this section we recall some of the concepts and facts used in the paper.

### 2.1 Graphical notions

We assume that the reader is familiar with basic graphical concepts from the area of graphical models. Their definitions can be found either in § 2.1.1 of (Lauritzen, 1996), in § 4.1 of (Cowell et al., 1999) or in § A.3 of (Studený, 2005). In particular, we have in mind the concepts of a chain graph, an undirected graph, an acyclic directed graph, an induced subgraph and the moral graph (of a graph). We will also need the concept of a *component*[1] of a chain graph and the concept of a *clique* of an undirected graph. The class of components of a chain graph $G$ will be denoted by $\mathcal{C}(G)$. The undirected version of a graph will be called its *underlying graph* in this paper. The set of *parents* of a set of nodes $C$ in a graph $G$ will be denoted by $pa_G(C)$. A *line* between nodes $a$ and $b$ will be denoted by $a — b$ in the text, an *arrow* from $a$ to $b$ by $a \to b$.

An important concept is that of a *decomposable* (undirected) graph, defined in § 2.1.2 of (Lauritzen, 1996), and the observation that its cliques can be ordered into a sequence $C_1, \ldots, C_m$, $m \geq 1$ satisfying the *running intersection property*:

$$\forall\, i \geq 2 \;\; \exists\, k < i \quad S_i \equiv C_i \cap (\bigcup_{j<i} C_j) \subseteq C_k\,. \quad (1)$$

[1] Some authors prefer to use a longer phrase a *chain component*.

It is a well-known fact that the collection of sets $S_i$, $2 \leq i \leq m$ does not depend on the choice of an ordering satisfying (1). We will call these sets *separators* of the graph. Moreover, the multiplicity $\nu(S)$ of a separator $S$, that is, the number of indices $i$ for which $S = S_i$ also does not depend on the choice of an ordering satisfying (1) – see Lemma 7.2 in (Studený, 2001; Studený, 2005).

Two types of configurations of three nodes in a graph will play an important role in the paper. An *immorality* in a chain graph $G$ is an induced subgraph for a set $T = \{a, b, c\}$, in which $a \rightarrow c \leftarrow b$ and $[a, b]$ is not an edge in $G$. A *flag* in $G$ is an induced subgraph, in which $a \rightarrow c$, $c — b$ and $[a, b]$ is not an edge.

Given two different chain graphs $G$ and $H$ with the same underlying graph, we say that $H$ is *larger* than $G$ if $a \rightarrow b$ in $H$ implies $a \rightarrow b$ in $G$. This means that $H$ has more lines than $G$. If $C$ is a component of a chain graph $G$ then by a *closure graph* for $C$ we will understand the moral graph of the induced subgraph for the set $D = C \cup pa_G(C)$, in notation $G_D$.

### 2.2 Bayesian networks

We understand a Bayesian network as a statistical model attributed to an acyclic directed graph. More specifically, given such a graph $G$ over a set of variables (= nodes) $N$ and a collection $\mathsf{X}_i$, $i \in N$ of individual sample spaces (i.e., non-empty finite sets) for variables, one can introduce the corresponding class of probability measures on $\prod_{i \in N} \mathsf{X}_i$, that is, a *statistical model*. The probability measures in the class can either be introduced as those which factorize recursively according to $G$ or, equivalently, as those which satisfy conditional independence restrictions given either by the moralization or by the d-separation criterion – see § 3.2.2 of (Lauritzen, 1996). In this paper, the phrase *Bayesian network model* will be used to name the above-mentioned statistical model. Thus, in learning Bayesian networks, the goal of a learning procedure is to determine that statistical model on basis of data.

*Remark.* Some authors define a Bayesian network as a pair: a graph and a probability measure which factorizes according to the graph. In our approach, we are interested in the whole class of probability measures which has the same structure, described solely by the graph. In brief, we are interested in **structure** of a Bayesian network. Thus, our interest in learning Bayesian networks is, therefore, restricted to *structural learning* – see Chapter 11 in (Cowell et al., 1999). For this reason we use the terms 'Bayesian network' and 'acyclic directed graph' as synonyms in the rest of the paper. Note that one can analogously ascribe a certain statistical model to every chain graph – see § 3.2.3 in (Lauritzen, 1996).

### 2.3 Equivalence and essential graph

Two Bayesian networks, respectively two chain graphs, are called *Markov equivalent* if they represent the same statistical model. In standard situations[2], this requirement is equivalent to the condition that they are *independence equivalent*, that is, they define the same collection of conditional independence restrictions. Verma and Pearl (1991) gave a direct graphical characterization of equivalent Bayesian networks: they are independence equivalent iff they have the same underlying graph and the same collection of immoralities. Note that Frydenberg (1990) achieved a similar result for chain graphs, but his characterization is more complex. On the other hand, the same equivalence characterization result as for Bayesian networks holds for chain graphs without flags – see Lemma 2 in (Studený, 2004a).

An equivalence class $\mathcal{G}$ of Bayesian networks (over $N$) can be described by its *essential graph* which is a (chain) graph $G^*$ defined as follows:

- $a \rightarrow b$ in $G^*$ iff $a \rightarrow b$ in $G$ for every $G \in \mathcal{G}$,
- $a — b$ in $G^*$ iff there are $G_1, G_2 \in \mathcal{G}$ such that $a \rightarrow b$ in $G_1$ and $b \rightarrow a$ in $G_2$.

EXAMPLE An example of the essential graph of an equivalence class is given in Figure 1.

A graphical characterization of essential graphs was presented by Andersson, Madigan

[2]See § 6.1.1 of (Studený, 2005) for an explanation.

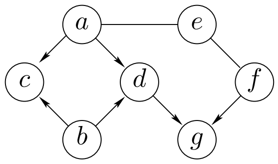


Figure 1: Example of an essential graph.

and Perlman (1997a). A consequence of their characterization is that every essential graph is a chain graph $G^*$ without flags such that, for every its component $C$, the induced subgraph $G^*_C$ is decomposable. Actually, it has been shown recently in (Studený, 2004a) and independently in (Roverato, 2005) that the essential graph of $\mathcal{G}$ is just the largest graph in the class of chain graphs without flags equivalent to the graphs in $\mathcal{G}$. Another useful fact is the following one – see Lemma 3 in (Studený, 2004a).

LEMMA 1 A chain graph $H$ without flags is equivalent to a Bayesian network iff, for every component $C$ of $H$, the induced subgraph $H_C$ is decomposable.

### 2.4 Imsets

An integer-valued function on the power set $\mathcal{P}(N) = \{A;\, A \subseteq N\}$ of $N$ will be called an *imset* over $N$. Of course, it can be viewed as a vector whose components are integers indexed by subsets of $N$. Given $A \subseteq N$ the symbol $\delta_A$ will denote an imset identifying this set:

$$\delta_A(B) = \begin{cases} 1 & \text{if } B = A, \\ 0 & \text{otherwise.} \end{cases}$$

*Remark.* In (Studený, 2001; Studený, 2005) a certain class of *structural imsets* is introduced. These imsets are used to describe probabilistic conditional independence structures. Every structural imset defines through an algebraic criterion a collection of conditional independence restrictions – the criterion is an analogy of graphical separation criteria mentioned in Section 2.2. Thus, a wider class of statistical models of conditional independence structure can be considered.

However, to describe Bayesian network models, it is suitable to consider only a certain subclass of the class of structural imsets. Given an acyclic directed graph $G$ over $N$, the *standard imset* for $G$ is given by the formula

$$u_G = \delta_N - \delta_\emptyset + \sum_{a \in N} \{\delta_{pa_G(a)} - \delta_{\{a\} \cup pa_G(a)}\}\,. \quad (2)$$

The point is that the standard imset is a uniquely determined representative of an equivalence class of Bayesian networks. The following result is proved as Corollary 7.1 in (Studený, 2001; Studený, 2005).

LEMMA 2 Two acyclic directed graphs $G$ and $H$ are independence equivalent iff $u_G = u_H$.

### 2.5 Quality criteria and imsets in learning Bayesian networks

It was shown in § 8.4.2 of (Studený, 2004c; Studený, 2005) that typical quality criteria used in practice have a special form. More specifically, every criterion $\mathcal{Q}(D,G)$ depending on data $D$ and a Bayesian network $G$ which is score-equivalent and decomposable has the form

$$\mathcal{Q}(D,G) = s^{\mathcal{Q}}_D + \sum_{S \subseteq N} t^{\mathcal{Q}}_D(S) \cdot u_G(S),$$

where $s^{\mathcal{Q}}_D$ is a constant depending on data and $[t^{\mathcal{Q}}_D(S)]_{S \subseteq N}$ is so-called *data vector* (relative to a criterion $\mathcal{Q}$).

EXAMPLE In the case of a well-known Bayesian information criterion (= BIC) one has the following formula for the respective data vector (Studený, 2004c; Studený, 2005):

$$\begin{aligned} t^{\mathsf{BIC}}_D(S) \;&=\; d \cdot H(\hat{P}_S | \prod_{i \in S} \hat{P}_i) \quad (3) \\ &-\frac{1}{2} \cdot \ln d \cdot \{\, |S| - 1 + \prod_{i \in S} r(i) - \sum_{i \in S} r(i)\,\}\,, \end{aligned}$$

where $\emptyset \neq S \subseteq N$, $d$ is the length of the database $D$, $H(*|*)$ denotes the relative entropy, $\hat{P}_S$ is the marginal of the empirical measure (based on $D$) for $S$ and $r(i) = |\mathsf{X}_i|$, $i \in S$ are the cardinalities of the respective individual sample spaces. Moreover, one has $t^{\mathsf{BIC}}_D(\emptyset) = 0$.

Thus, if the criterion and data are fixed, it is a sum of a constant and a linear function of the standard imset. Moreover, a move from a Bayesian network to its inclusion neighbour can be described by a certain *differential imset* which has an elementary form

$$u_{\langle a,b|C\rangle} = \delta_{\{a,b\}\cup C} + \delta_C - \delta_{\{a\}\cup C} - \delta_{\{b\}\cup C},$$

where $a \neq b \in N$, $C \subseteq N \setminus \{a,b\}$. Thus, it can be interpreted in terms of conditional independence $a \perp\!\!\!\perp b \,|\, C$. In particular, the respective change in the value of $\mathcal{Q}$ takes a neat form of the scalar product of two vectors:

$$\langle t_D^{\mathcal{Q}}, u_{\langle a,b|C\rangle}\rangle = \sum_{S\subseteq N} t_D^{\mathcal{Q}}(S) \cdot u_{\langle a,b|C\rangle}(S)\,.$$

## 3 General formula

In this section we give a formula for the standard imset on basis of any chain graph $H$ over $N$ which is **equivalent to a Bayesian network**.

It is a well-known fact that all closure graphs $\bar{H}(C) \equiv (H_{C\cup pa_H(C)})^{mor}$ for components $C \in \mathcal{C}(H)$ of such a graph are decomposable (Andersson et al., 1997b). Let $\bar{\mathcal{K}}(C)$ denote the collection of cliques of $\bar{H}(C)$ and $\bar{\mathcal{S}}(C)$ the collection of separators in $\bar{H}(C)$. Further, let $\bar{\nu}_C(S)$ denote the multiplicity of a separator $S$ in $\bar{H}(C)$.

The *standard imset* for $H$ is given by the following formula:

$$\begin{aligned} u_H \;\; = \;\; & \delta_N - \delta_\emptyset + \sum_{C\in\mathcal{C}(H)} \{\, \delta_{pa_H(C)} \qquad (4) \\ & - \sum_{K\in\bar{\mathcal{K}}(C)} \delta_K + \sum_{S\in\bar{\mathcal{S}}(C)} \bar{\nu}_C(S)\cdot\delta_S \,\}\,. \end{aligned}$$

The point is that the formula (4) gives the same result for equivalent chain graphs.

Lemma 3 Let $G$ and $H$ are equivalent chain graphs such that there exists a Bayesian network equivalent to them. Then $u_G = u_H$.

The proof of Lemma 3 is omitted for it is too long. It is based on the idea of a certain elementary operation of 'feasible' merging of components in a chain graph. This operation gives an equivalent chain graph which is, however, larger. Thus, every chain graph can successively be transformed into the respective largest chain graph. Moreover, the formula (4) gives the same result after each application of that elementary operation.

| subset of $N$ | value |
|---|---|
| $\{a,b,c,d,e,f,g\}$ | $+1$ |
| $\emptyset$ | $+1$ |
| $\{a,b\}$ | $+2$ |
| $\{a,b,c\}$ | $-1$ |
| $\{a,b,d\}$ | $-1$ |
| $\{a,e\}$ | $-1$ |
| $\{b\}$ | $-1$ |
| $\{d,f\}$ | $+1$ |
| $\{d,f,g\}$ | $-1$ |
| $\{e\}$ | $+1$ |
| $\{e,f\}$ | $-1$ |

Table 1: The standard imset for the essential graph from Figure 1. The values for remaining subsets of $N$ are zero.

Of course, if $H = G$ is a Bayesian network, then (4) gives the same result as (2).

Corollary 1 Let $G$ be a Bayesian network and $H$ the essential graph of the respective equivalence class. Then the formula (4) gives the standard imset for $G$.

Example In Table 1 we give the standard imset for the essential graph from Figure 1.

## 4 Graphical characteristics

The formula (4) can be simplified for chain graphs without flags. In this section, we introduce some characteristics of these graphs that will be used in the simplified formula given in Section 5.

### 4.1 Initial components

A component $C \in \mathcal{C}(H)$ of a chain graph $H$ such that $pa_H(C) = \emptyset$ will be called an *initial component* of $H$. Actually, it is a counterpart of the concept of a terminal component. Let us denote by $i(H)$ the number of initial components in $H$. Note that this number appears to be the same for equivalent chain graphs.

Example The chain graph $H$ in Figure 1 has two initial components: $\{b\}$ and $\{a,e,f\}$. Thus, $i(H) = 2$.

### 4.2 Core

We will say that a set $B$ of nodes in a chain graph $H$ over $N$ is *idle* if the following two conditions hold:

- $\forall\, b_1 \neq b_2 \in B,\ \ [b_1, b_2]$ is an edge in $H$,
- $\forall\, a \in N \setminus B, \forall\, b \in B \quad a \to b$ in $H$.

The meaning of these conditions is that no non-trivial conditional independence statement represented in $H$ involves variables in $B$. The second condition implies that every component of $H$ intersecting an idle set $B$ is contained in $B$.

One can easily show that every chain graph $H$ over $N$ has a unique maximal idle set of nodes, possibly empty. The set can be shown to be the same for equivalent chain graphs. Note that if $H$ is the essential graph of an equivalence class of Bayesian networks and has a non-empty maximal idle set, then the maximal idle set is a component of $H$. Actually, it is a *superterminal component* of $H$, that is, a non-empty set $C \subseteq N$ such that $H_C$ is a complete undirected graph and $pa_H(c) = N \setminus C$ for every $c \in C$.

Let us call the complement of the maximal idle set in $H$ the *core* of $H$ and denote it by $core(H)$. The class of *core-components*, that is, components of $H$ contained in the core, will be denoted by $\mathcal{C}_{core}(H)$. Observe that if the core is non-empty then every initial component is a core-component.

### 4.3 Cliques and separators

If $H$ is a chain graph equivalent to a Bayesian network then every its **core-component** $C$ induces a decomposable graph $H_C$ by Lemma 1. Let us denote by $\mathcal{K}(C)$ the class of its cliques, by $\mathcal{S}(C)$ the collection of its separators, and by $\nu_C(S)$ the multiplicity of $S \in \mathcal{S}(C)$ in $H_C$. Note that, since $C$ is connected, every $S \in \mathcal{S}(C)$ is a **non-empty** proper subset of $core(H)$.

Example The chain graph $H$ from Figure 1 has an empty maximal idle set, i.e., the core is $core(H) = N$. Its core-components are $C_1 = \{a, e, f\}$, $C_2 = \{b\}$, $C_3 = \{c\}$, $C_4 = \{d\}$ and $C_5 = \{g\}$. All components except for $C_1$ have only one clique and no separator. The set of cliques of $H_{C_1}$ is $\mathcal{K}(C_1) = \{\{a, e\}, \{e, f\}\}$ and the set of its separators is $\mathcal{S}(C_1) = \{\{e\}\}$.

On the other hand, if the example is modified by incorporating a new node $h$ such that $n \to h$ for every $n \in \{a, \dots, g\}$ then the maximal idle set would be $\{h\}$.

### 4.4 Parent sets

A set $P \subseteq N$ will be called a *parent set* in a chain graph $H$ if it is **non-empty** and there exists a **core-component** $C \in \mathcal{C}_{core}(H)$ with $P = pa_H(C)$. The *multiplicity* $\tau(P)$ of a parent set $P$ is the number of $C \in \mathcal{C}_{core}(H)$ with $P = pa_H(C)$. Let us denote the collection of parent sets in $H$ by $\mathcal{P}_{core}(H)$. Evidently, every $P \in \mathcal{P}_{core}(H)$ is a proper subset of $core(H)$.

Example The parent set of the chain graph $H$ from Figure 1 is $\mathcal{P}_{core}(H) = \{\{a, b\}, \{d, f\}\}$. The multiplicities are $\tau(\{a, b\}) = 2$ and $\tau(\{d, f\}) = 1$.

## 5 Adapted formula

Of course, the characteristics introduced in Sections 4.3 and 4.4 are not invariants of equivalence of chain graphs. Nevertheless, it follows from the formula below that they allow one to introduce some invariants.

Lemma 4 Let $H$ be a chain graph without flags equivalent to a Bayesian network with $core(H) \neq \emptyset$. Then the standard imset for $H$ is given by

$$\begin{aligned} u_H &= \delta_{\mathrm{core}(H)} - \sum_{C \in \mathcal{C}_{core}(H)} \sum_{K \in \mathcal{K}(C)} \delta_{K \cup \mathrm{pa}_H(C)} \\ &+ \sum_{C \in \mathcal{C}_{core}(H)} \sum_{S \in \mathcal{S}(C)} \nu_C(S) \cdot \delta_{S \cup \mathrm{pa}_H(C)} \\ &+ \sum_{P \in \mathcal{P}_{core}(H)} \tau(P) \cdot \delta_P + \{i(H) - 1\} \cdot \delta_\emptyset \,. \end{aligned}$$

Although the formula is not difficult to verify we omit its proof to keep the reasonable length of the paper. The formula implies that the class of sets

$$\{K \cup pa_H(C); K \in \mathcal{K}(C), C \in \mathcal{C}_{core}(H)\}$$

and the class of sets

$$\mathcal{P}_{core}(H) \cup \{S \cup pa_H(C); S \in \mathcal{S}(C), C \in \mathcal{C}_{core}(H)\}$$

are invariants of equivalence of considered chain graphs. The point is that, in the case of a non-trivial essential graph, these classes of sets are disjoint. We omit the proof of the following technical lemma.

LEMMA 5 Let $H$ be the essential graph of an equivalence class of Bayesian networks over $N$ such that $u_H \neq 0$. Then, for every $L \subseteq N$, **exclusively** one of the following options occurs:

**(a)** $L = core(H)$ and $u_H(L) = +1$,

**(b)** $L = K \cup pa_H(C)$ for $K \in \mathcal{K}(C)$, $C \in \mathcal{C}_{core}(H)$ and $u_H(L) = -1$,

**(c)** $L = S \cup pa_H(C)$ for $S \in \mathcal{S}(C)$, $C \in \mathcal{C}_{core}(H)$ and $u_H(L) = \nu_C(S) > 0$,

**(d)** $L = P$ for $P \in \mathcal{P}_{core}(H)$ and $u_H(L) = \tau(P) > 0$,

**(e)** $L = \emptyset$ and $u_H(L) = i(H) - 1$,

**(f)** none of above cases occurs and $u_H(L) = 0$.

It follows from Lemma 5 that, given a non-zero standard imset, one can determine simply the core of the essential graph, the number of its initial components, the collection of sets from **(b)** and the union of sets from **(c)** and **(d)**.

## 6 Reconstruction algorithm

Lemma 5 is a basis of a two-stage reconstruction algorithm for the essential graph from the standard imset; the proof of its correctness is omitted for it requires several additional technical lemmas.

The first stage of the algorithm is described in Table 2. The output of this stage is an ordered sequence of sets $\mathcal{T} = \{T_i\}_{i=1}^n$ where $\emptyset \neq T_i \subseteq N$, $i = 1, \ldots, n$.

The basic idea of the first stage of the algorithm is to search for a clique $V = K \cup pa_H(C)$, $K \in \mathcal{K}(C)$ of the closure graph for a component $C \in \mathcal{C}(H)$ that is a leaf of a junction tree for cliques of $\bar{H}(C)$. The class $\mathcal{V}$ (line 13) is the class of sets that are candidates for $V$ since their imset values are negative. The set $W$ (line 14) contains variables that appear in at least two sets from $\mathcal{V}$. This is a necessary condition for the variables belonging to a separator $Z = S \cup pa_H(C)$, $S \in \mathcal{S}(C)$ in a junction tree for cliques of $\bar{H}(C)$. Therefore, $V \cap W$ is a candidate for the separator $Z$ attributed to the clique $V$. Moreover, $V$ is required to contain at least one variable that is not in $Z$ and if $Z$ is non-empty then it must have a positive imset value (line 19).

Once such a set $V$ is found, it is appended to $\mathcal{T}$ (line 20), its corresponding semi-elementary imset

$$u_{\langle Q,R|Z\rangle} \;=\; \delta_{Q\cup R\cup Z} + \delta_Z - \delta_{Q\cup Z} - \delta_{R\cup Z}$$

is subtracted from the input imset (line 22) and the set $Y$ is shrunked (line 23). Note that the assumption that the input is the standard imset for an acyclic directed graph ensures that the procedure will not reach a dead end.

EXAMPLE The ordered sequence of subsets that is the outcome of the first stage for the imset given in Table 1 is

$$\{a,b,c\}, \{d,f,g\}, \{a,b,d\}, \{a,e\}, \{b\}, \{e,f\}\,. \quad (5)$$

The second stage of the reconstruction algorithm is presented in Table 3. The sequence of sets $\mathcal{T} = \{T_i\}_{i=1}^n$ is used to get the essential graph $H$ corresponding to the imset that was input of the first stage.

Elements of the sequence $\mathcal{T}$ are taken in the reverse order. Each set $T_j$ corresponds to a clique $K$ of the induced subgraph $H_C$ for a component $C$ of $H$, i.e. $T_j = K \cup pa_H(C)$, $K \in \mathcal{K}(C)$, $C \in \mathcal{C}(H)$. Therefore, $T_j$ is partitioned into three subsets: $R$, $X$ and $Z \setminus X$ (lines 7-14). More specifically,

- if $X$ is non-empty then it is a separator in $H_C$ attributed to the clique $K$,
- $R$ is the residual $K \setminus X$ of the clique $K$, and
- $Z \setminus X$ is the parent set of the respective component $C$.

Thus, edges are added to the constructed essential graph $H$ as follows: lines join all pairs of distinct nodes in $X \times R$ and $R \times R$, while arrows are directed from $Z \setminus X$ to $R$ (lines 15-20).

EXAMPLE Of course, the output of the second stage of the algorithm applied to the ordered sequence (5) of subsets of $N$ is the essential graph in Figure 1.

Table 2: The first stage of the reconstruction algorithm.

| | |
|---|---|
| Input: | a standard imset $u$ over a non-empty set of variables $N$ |
| Output: | an ordered sequence $\mathcal{T} = \{T_i\}_{i=1}^{n}$ of cliques $\emptyset \neq T_i \subseteq N, i = 1, \ldots, n$ of closure graphs for components of the essential graph corresponding to the imset $u$ |

```
Y := N; T := ∅;
repeat
    if u = 0 then
        T := append(T, Y);
        goto 25;
    else
        if u(Y) = 0 then
            T := append(T, Y);
            M := {M ⊆ Y : u(M) ≠ 0};
            Y := arg max_{M ∈ M} |M|;
            goto 3;
        else
            V := {V ⊆ Y : u(V) < 0};
            W := {w ∈ Y : |{V ∈ V : w ∈ V}| ≥ 2};
            L := {L ⊆ Y : u(L) > 0 & L ≠ Y} ∪ {∅};
            for V ∈ V do
                Z := V ∩ W;
                R := V \ Z;
                if R ≠ ∅ & Z ∈ L then
                    T := append(T, V);
                    Q := Y \ V;
                    u := u − u_⟨Q,R|Z⟩;
                    Y := Y \ R;
                    goto 3;
return T;
```

Table 3: The second stage of the reconstruction algorithm.

| | |
|---|---|
| Input: | an ordered sequence $\mathcal{T} = \{T_i\}_{i=1}^{n}$ of cliques $\emptyset \neq T_i \subseteq N, i = 1, \ldots, n$ of closure graphs for components of the essential graph corresponding to the imset $u$ |
| Output: | the essential graph $H$ over $N$ with a set of edges $\mathcal{E}$ (the one which corresponds to $u$) |

```
E := ∅;
Y := T_n;
for (u, v) ∈ Y × Y do
    if u ≠ v then E := E ∪ {u — v};
H := the graph over Y with edges E;
for j ∈ {n − 1, ..., 1} do
    Z := T_j ∩ Y;
    R := T_j \ Z;
    X := ∅;
    if Z ≠ ∅ then
        G := the graph over Z with edges E ∩ (Z × Z);
        for D ∈ C(G) do
            if {u — v; u ≠ v, u, v ∈ D} ⊆ E & ∀ d ∈ D : pa_H(d) = pa_G(d) = Z \ D then
                X := D;
    for (u, v) ∈ X × R do
        E := E ∪ {u — v};
    for (u, v) ∈ (Z \ X) × R do
        E := E ∪ {u → v};
    for (u, v) ∈ R × R do
        if u ≠ v then E := E ∪ {u — v};
    Y := Y ∪ T_j;
    H := the graph over Y with edges E;
return H;
```

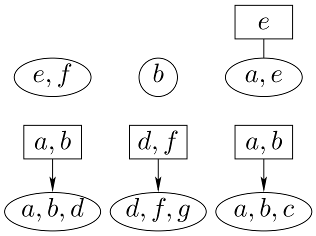


Figure 2: Units of a hierarchical junction tree.

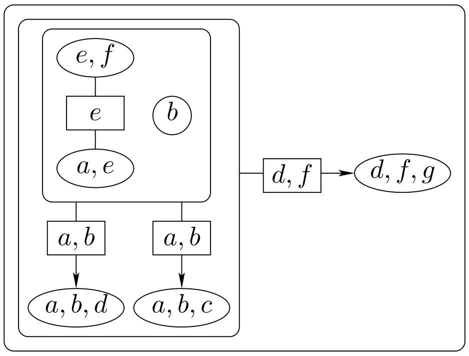


Figure 3: A hierarchical junction tree.

## 7 Construction of a hierarchical junction tree

The sequence of sets that is the outcome of the first stage (Table 2) can also be used to construct a hierarchical junction tree similar to those introduced in (Puch et al., 2004).

Each set in the sequence defines a node of the hierarchical junction tree whose entering edges could be labelled by sets $X$ or $Z$ obtained during the second stage of the reconstruction algorithm (see Table 3). More specifically, each node $T_j$ may or may not be ascribed an entering edge and the edge can be labelled either by a separator $X$ (if $X \neq \emptyset$) or by a parent set $Z$ (if $X = \emptyset$ and $Z \equiv Z \setminus X \neq \emptyset$). These units can be used then to compose the whole hierarchical junction tree. Due to the lack of space we omit details of the construction and give two examples instead.

Example The nodes of a hierarchical junction tree constructed from the sequence (5) are given in Figure 2, including their attributed separators and parent sets. The resulting hierarchical junction tree is given in Figure 3.

Example Figure 4 gives another example of

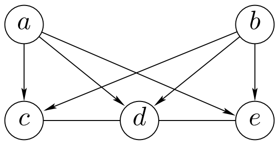


Figure 4: An essential graph.

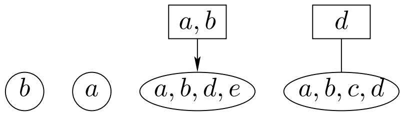


Figure 5: Units of a hierarchical junction tree.

an essential graph $H$. The first stage of the reconstruction algorithm applied to the standard imset $u_H$ ends with the sequence of sets

$$\{a,b,c,d\},\ \{a,b,d,e\},\ \{a\},\ \{b\}\,.$$

The nodes of the respective hierarchical junction tree are given in Figure 5, including their attributed separators and parent sets. The resulting hierarchical junction tree is given in Figure 6. In that picture, the parent set $pa_H(\{c,d,e\}) = \{a,b\}$ is attributed to just one node of the hierarchical junction tree. One can perhaps draw a picture in which every parent set is ascribed to every node of the respective component of the hierarchical junction tree.

## Conclusions

The presented procedures for the transition between graphical and algebraic representatives of a Bayesian network model can be the first step on the way towards a fully algebraic method for learning Bayesian networks. We hope that the procedures can be utilized to find a characterization of the inclusion neighbourhood of a given

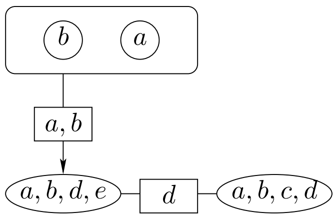


Figure 6: A hierarchical junction tree.

Bayesian network model in terms of the standard imset. This will be a topic of a future research. We also plan to study the polytope generated by standard imsets over $N$ hoping that linear programming maximization methods can be applied in learning Bayesian networks.


### Acknowledgments

This research has been supported by the grant GAČR nr. 201/04/0393.


## References


S. A. Andersson, D. Madigan, and M. D. Perlman. 1997a. A characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25:505–541.

S. A. Andersson, D. Madigan, and M. D. Perlman. 1997b. On the Markov equivalence of chain graphs, undirected graphs and acyclic digraphs. *Scandinavian Journal of Statistics*, 24:81–102.

V. Auvrey and L. Wehenkel. 2002. On the construction of the inclusion boundary neighbourhood for Markov equivalence classes of Bayesian network structures. In A. Darwiche and N. Friedman, editors, *Uncertainty in Artificial Intelligence 18*, pages 26–35. Morgan Kaufmann.

R. R. Bouckaert. 1995. *Bayesian belief networks: from construction to inference*. Ph.D. thesis, University of Utrecht.

R. Castelo. 2002. *The discrete acyclic digraph Markov model in data mining*. Ph.D. thesis, University of Utrecht.

D. M. Chickering. 2002. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554.

R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and Spiegelhalter D. J. 1999. *Probabilistic Networks and Expert Systems*. Springer-Verlag.

M. Frydenberg. 1990. The chain graph Markov property. *Scandinavian Journal of Statistics*, 17:333–353.

T. Kočka and R. Castelo. 2001. Improved learning of Bayesian networks. In J. Breese and D. Koller, editors, *Uncertainty in Artificial Intelligence 17*, pages 269–276. Morgan Kaufmann.

S. L. Lauritzen. 1996. *Graphical Models*. Clarendon Press.

C. Meek. 1995. Causal inference and causal explanation with background knowledge. In P. Besnard and S. Hanks, editors, *Uncertainty in Artificial Intelligence 11*, pages 403–410. Morgan Kaufmann.

R. O. Puch, J. Q. Smith, and C. Bielza. 2004. Hierarchical junction trees: conditional independence preservation and forecasting in dynamic Bayesian networks with heterogeneous evolution. In J. A. Gámez, S. Moral, and A. Salmerón, editors, *Advances in Bayesian Networks*, pages 57–75. Springer-Verlag.

R Development Core Team, 2004. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3.

A. Roverato. 2005. A unified approach to the characterization of equivalence classes for DAGs, chain graphs with no flags and chain graphs. *Scandinavian Journal of Statistics*. Submitted.

M. Studený. 2001. *On mathematical description of probabilistic conditional independence structures.* Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic. Thesis for DrSc degree, available at `ftp://ftp.utia.cas.cz/pub/staff/studeny/msdoktor.ps`.

M. Studený. 2004a. Characterization of essential graphs by means of the operation of legal merging of components. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12:43–62.

M. Studený. 2004b. Characterization of inclusion neighbourhood in terms of the essential graph. *International Journal of Approximate Reasoning.* To appear.

M. Studený. 2004c. Learning. A preliminary version of Chapter 8 of (Studený 2005) available at `ftp://ftp.utia.cas.cz/pub/staff/studeny/chap8.ps`.

M. Studený. 2005. *On Probabilistic Conditional Independence Structures.* Springer-Verlag. To be published in October 2004 (copyright 2005).

T. Verma and J. Pearl. 1991. Equivalence and synthesis of causal models. In P. P. Bonissone, M. Henrion, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 6*, pages 220–227. Elsevier.