On the inclusion problem*

Tomáš Kočka
Univ. of Economics Prague
kocka@vse.cz

Remco R. Bouckaert
Xtal Mountain Inf. Tech.
rrb@xm.co.nz

Milan Studený
Acad. Sci. Czech Rep.
studeny@utia.cas.cz

February 14, 2001

Abstract

Every directed acyclic graph (DAG) over a finite non-empty set of variables (= nodes) N induces an independence model over N, which is a list of conditional independence statements over N. The *inclusion problem* is how to characterize (in graphical terms) whether all independence statements in the model induced by a DAG K are in the model induced by a second DAG L. Meek [8] conjectured that this inclusion holds iff there exists a sequence of DAGs from K to L such that only arrow removal and 'legal' arrow reversal operations are performed to get the next DAG in the sequence.

In this report we give various characterizations of inclusion of DAG models and the proof of Meek's conjecture in the case that the DAGs K and L differ in at most one adjacency. As a warming up a rigorous proof of well-known graphical characterizations of equivalence of DAGs, which is a highly related problem, is given. Furthermore, we give intuition how to characterize inclusion of DAG models in general and describe possible strategies how to verify Meek's conjecture even if the DAGs K and L differ in more than one edge.

1 Introduction

Learning Bayesian network structures requires search in the space of directed acyclic graphs (DAGs). To prove that such learning algorithms return (local) optimal networks, the search space needs to be characterized. A natural way in doing this is to consider the set of conditional independence statements represented by the DAGs in the search space. Once it is known how to characterize all the properties of two DAGs K and L such that independence statements represented in K are represented in L as well, efficient search algorithms can be designed based on this characterization. This characterization problem is called the *inclusion problem*.

Chris Meek [8] formulated a conjecture which says that this is if and only if a special sequence of DAGs $H_1, \ldots, H_n$ starting with $K = H_1$ and ending in $L = H_n$ exists. Further, H_{i+1} is obtained from H_i either by removal of an arrow or by performing a single arrow reversal (this arrow reversal is special in that it does not introduce new represented independence statements). The conjecture claims that if all independence statements represented in K are represented in L as well, then there exists such a sequence. Many search algorithms for learning Bayesian

*This research has been supported by the grant MŠMT n. VS96008, by the grant GAČR n. 201/01/1482 and by the grant FRVŠ n. 2001/1433. The authors had an opportunity to discuss the topic of this paper during the seminar "Conditional independence structures" held at the Fields Institute, University of Toronto in October 1999 and during HSSS research kitchen "Learning conditional independence models" held in Třešť in October 2000.

networks rely on the conjecture being true for optimality of the learned network structures [3, 2].

The aim of this report is to give an overview of current state of our research in the inclusion problem. Since the inclusion problem seems to be difficult (some people claim that it is a very difficult problem) it is useful to have an overview of the attempts that have been done to get a good sense of intuition for desired future solving of the problem. To that extent, this report views the problem from different angles and lists preliminary results for the simple case when the DAGs K and L differ in only one adjacency.

The next section describes the basic concepts and notation used in the rest of the report. In Section 3 equivalence of two DAGs (i.e. coincidence of their induced independence models) is studied. In Section 4 we formulate the inclusion problem; in Section 5 we give an overview of some necessary conditions. Section 6 contains the main result: we characterize the case when two DAGs differ in only one adjacency. In Section 7 we formulate conjectures for the general case. We end with conclusions and directions of future research in Section 8.

2 Basic concepts

Throughout the paper the symbol N denotes a non-empty finite set of variables which are identified with nodes of graphs. Lower case letters like a, b, c, d, t, u, v, w will be used to denote elements of N while capital letters like A, B, C, S, P, T, U, V, W, X, Y, Z will be used to denote subsets of N. Occasionally, we write only u to denote a singleton $\{u\}$ for $u \in N$ and use juxtaposition AB to denote the union $A \cup B$ of (usually disjoint) subsets of N. Inclusion of sets will be indicated by the symbol $\subseteq$ while $\subset$ is reserved for strict inclusion (i.e. $A \subset B$ means that $A \subseteq B$ and $A \neq B$). Symbol $|A|$ denotes the number of elements of a finite set A. Possibly indexed capital letters K, L, G, H will be used to denote graphs having N as the set of nodes. These graphs can be used to describe certain conditional (in)dependence structures over N (in the way specified in Section 2.2).

Independence and dependence statements over N correspond to special *disjoint triplets* over N. The symbol $\langle A, B|C\rangle$ denotes a triplet of pairwise disjoint subsets A, B, C of N. This notation anticipates the intended meaning: the set of variables A is conditionally independent or dependent of the set of variables B given the set of variables C. That is why the third set C is separated by a straight line: it has a special meaning of the conditioning set. The symbol $\mathcal{T}(N)$ will denote the class of all disjoint triplets over N:

$$\mathcal{T}(N) = \{\langle A, B|C\rangle\ ;\ A, B, C \subseteq N \quad A \cap B = B \cap C = A \cap C = \emptyset\}.$$

2.1 Graphical concepts

A *directed graph* G over a set of *nodes* N is specified by a collection of *arrows*, that is a collection $\mathcal{A}(G)$ of ordered pairs (u, v) of distinct nodes $u, v \in N$, $u \neq v$. We write $u \to v$ in G or shortly $u \to v$ $[G]$ to denote that $(u, v) \in \mathcal{A}(G)$; the symbol of the graph can be omitted if it is clear from the context which graph is indicated. An analogous principle in notation is used in pictures throughout the report. In an arrow $u \to v$, denoted alternatively by $v \leftarrow u$, u is called the *head node* or the *head* and v the *tail node* or the *tail*. Furthermore, we say that u is the *parent* of v and v is the *child* of u. The set of parents of u in G will be denoted by $pa_G(u)$, the set of children by $ch_G(u)$. A *subgraph* of a directed graph G over N is determined by a non-empty set of its nodes $A \subseteq N$ and by the set of its arrows which is a subset of $\mathcal{A}(G) \cap (A \times A)$ (strict inclusion is allowed). The *induced subgraph* of G for a non-empty set $B \subseteq N$ is a graph G_B over B having $\mathcal{A}(G_B) = \mathcal{A}(G) \cap (B \times B)$ as the collection of arrows.

We write $u \leftrightarrow v\ [G]$ to denote that there is an *edge* or an adjacency between nodes u and v in G which means that either $u \to v$ in G or $u \leftarrow v$ in G (note that simultaneous occurence of $u \to v$ and $u \leftarrow v$ is allowed in a general directed graph). The set of edges in a directed graph G is the collection of two-element subsets of N:

$$\mathcal{E}(G) \;=\; \{\; \{u,v\}\;;\; u \leftrightarrow v\ [G]\;\}\,.$$

If there is no edge between u and v in G then we write $u \not\leftrightarrow v\ [G]$ to denote this *non-adjacency.*

A *trail* in G (between nodes u and v) is a sequence of (not necessarily distinct) nodes $w_1, \ldots, w_k$, $k \geq 1$ such that $w_i \leftrightarrow w_{i+1}\ [G]$ for every $1 \leq i < k$ (and either $w_1 = u$, $w_k = v$ or $w_1 = v$, $w_k = u$). It is called a *path* if all nodes $w_1, \ldots, w_k$ are distinct. A *section* of a path $w_1, \ldots, w_k$, $k \geq 1$ is a path $w_i, \ldots, w_j$ where $1 \leq i \leq j \leq k$.

The trail (path) is called *directed* if $w_i \to w_{i+1}\ [G]$ for $i = 1, \ldots, k-1$. We say that it is a path from a node u to a node v (from $A \subseteq N$ to $B \subseteq N$) if $w_1 = u$ and $w_k = v$ ($w_1 \in A$ and $w_k \in B$). A node u is called an *ancestor* of a node v in G (or alternatively v is a *descendant* of u in G) if there is a directed path from u to v in G. Observe that every node is its own ancestor and its own descendant as paths with only a single node are regarded as paths. The symbol $an_G(A)$ will denote the set of all ancestors of nodes of a set $A \subseteq N$ in G and $ds_G(u)$ the set of descendants of a node u in G. A *terminal node* is a node without children (equivalently without distinct descendants).

A *directed cycle* is a directed trail $w_1, \ldots, w_k$, $k \geq 3$ such that $w_1 = w_k$ and $w_1, \ldots, w_{k-1}$ are distinct nodes. A *directed acyclic graph* (DAG) is a directed graph without directed cycles. This phrase is widely accepted despite the fact that grammatical rules require to use the term 'acyclic directed graph'. Observe that arrows $u \to v$ and $u \leftarrow v$ cannot occur simultaneously in a DAG (otherwise $u \to v \to u$ is a directed cycle). Thus, every trail (path) in a DAG, which is defined as a sequence of nodes, has uniquely determined the (type of) arrows connecting consecutive nodes. Another consequence is that $|\mathcal{A}(G)| = |\mathcal{E}(G)|$ for every DAG G. Another observation is that a subgraph of a DAG is also a DAG. A well-known equivalent definition of a DAG is as follows: it is a directed graph and all nodes can be ordered into a sequence $u_1, \ldots, u_n$ such that $pa_G(u_i) \subseteq \{u_j\,; 1 \leq j < i\}$ for every $i = 1, \ldots, n$. This ordering is called *causal ordering* for G. Note that every DAG has at least one terminal node.

An *undirected graph* H over N is specified by a collection $\mathcal{L}(H)$ of two-element subsets of N, which are called *lines* in H. H is called *complete* if every two-element subset of N is a line in H. A *path* in an undirected graph H (and related concepts) is defined analogously to the directed case: it is a sequence of distinct nodes $w_1, \ldots, w_k$, $k \geq 1$ such that $\{w_i, w_{i+1}\} \in \mathcal{L}(H)$ for every $1 \leq i < k$. By the *underlying graph* of a directed graph G over N is understood an undirected graph H for which $\mathcal{L}(H) = \mathcal{E}(G)$. We say that distinct nodes u, v, w form an *immorality* in a directed graph G and write $(u,v) \rightsquigarrow w\ [G]$ if $u \to w$ in G, $v \to w$ in G and $u \not\leftrightarrow v\ [G]$. In fact, an immorality in a DAG G is nothing but a special induced subgraph of G.

2.2 Induced models

One possible ways of associating independence models with DAGs is by the d-separation criterion from [9]. Let $\pi : w_1, \ldots, w_k$, $k \geq 1$ be a path in a DAG G. We say that w_i, $1 < i < k$ is a *collider node* of π if $w_{i-1} \to w_i$ in G and $w_i \leftarrow w_{i+1}$ in G. Every other node of π is called a *non-collider node* of π. A path π in G is *active* with respect to a set $C \subseteq N$ (shortly w.r.t. C) if

- every non-collider node of π is outside C,
- every collider node of π has a descendant in C.

Suppose $\langle A,B|C\rangle \in \mathcal{T}(N)$ is a disjoint triplet over N, one says that A and B are *d-connected* given C in a DAG G, written $A \not\perp\!\!\!\perp B \,|\, C\ [G]$, if there exists a path between a node $a \in A$ and a node $b \in B$ in G which is active w.r.t. C. In the opposite case one says that A and B are *d-separated* by C in G, written $A \perp\!\!\!\perp B \,|\, C\ [G]$. We also say that $\langle A,B|C\rangle$ is *represented* in G according to the d-separation criterion. The induced independence model $\mathcal{I}(G)$ and the induced dependence model $\mathcal{D}(G)$ are defined as follows:

$$\mathcal{I}(G) = \{\, \langle A,B|C\rangle \in \mathcal{T}(N)\,; \quad A \perp\!\!\!\perp B \,|\, C\ [G]\,\},$$

$$\mathcal{D}(G) = \{\, \langle A,B|C\rangle \in \mathcal{T}(N)\,; \quad A \not\perp\!\!\!\perp B \,|\, C\ [G]\,\}.$$

An alternative to the d-separation criterion is the moralization criterion [6]. The *moral graph* G^{mor} of a DAG G over N is an undirected graph over N which has the following set of lines:

$$\mathcal{L}(G^{mor}) \;=\; \{\; \{u,v\}\;;\;\; u \leftrightarrow v\ [G] \;\;\text{or}\;\; (u,v) \rightsquigarrow w\ [G] \text{ for some } w \in N\,\}\,.$$

That means, edges are added to connect all unadjacent nodes having a common child (i.e. parents are 'married' so that immoralities are removed in this way) and the underlying graph of the resulting graph is taken. Testing whether $\langle A,B|C\rangle \in \mathcal{T}(N)$ is represented in a DAG G over N according to the moralization criterion is a stepwise procedure. First, one takes the induced subgraph G_D where $D = an_G(ABC)$. Second, the moral graph H of G_D is made. Third, one checks whether every path in H from A to B contains a node in C (i.e. whether C separates between A and B in H). If so, $\langle A,B|C\rangle$ is represented in G according to the moralization criterion. It is well-known fact that this occurs iff A and B are d-separated by C in G [6].

The next modification (strengthening) of d-separation will be used in later proofs and is essential for formulation of Conjecture 3. Let G be DAG over N, $C \subseteq N$ and $a,b \in N \setminus C$ are distinct nodes. Let $\pi : w_1,\ldots,w_k$, $k \geq 2$ be a path in G between $a = w_1$ and $b = w_k$ which is active w.r.t. C. Every collider d of π which is not in C has necessarily a descendant $c \in C$, $c \neq d$ in G. By a *rope* for d (with respect to π) will be understood a directed path $\rho : t_1,\ldots,t_r$, $r \geq 2$ in G from $d = t_1$ to a node $c = t_r$ in C such that

- ρ is outside C with exception of c, i.e. $t_1,\ldots,t_{r-1} \notin C$,
- ρ does not share a node with π except d, i.e. $t_2,\ldots,t_r \notin \{w_1,\ldots,w_k\}$.

Let us denote by $col(\pi, C)$ the set of collider nodes of π which are outside C.

A *dependence complex* (between a and b) for C in G is a special subgraph κ of G. First, we specify the collection of arrows of a dependence complex. A complex κ is specified by the following items

- a path π in G which is active w.r.t. C,
- a collection of ropes $\{\rho(d)\,;\, d \in col(\pi, C)\,\}$ with respect to π,

where every collider $d \in col(\pi, C)$ has assigned only one rope $\rho(d)$ in κ and the ropes for distinct colliders do not share a node. The collection of arrows in κ then consist of the arrows involved in π and $\rho(d)$ for $d \in col(\pi, C)$. Second, we specify the set of nodes of a dependence complex as the set of head nodes and tail nodes of the chosen arrows. Thus, κ is a subgraph of G which need not have whole N as the set of nodes. Instead of dependence complex for C in G, we say or shortly *C-complex* in G (between A and B in case $a \in A$ and $b \in B$).

Let us emphasize that every dependence complex κ uniquely decomposes into the path π and the collection of ropes. Indeed, every node of a given subgraph κ of G, which was constructed as a dependence complex in G for a set $C \subseteq N$ and $a, b \in N \setminus C$, can be classified into one of three groups according to the number of edges of κ 'entering' the node (this number varies from 1 to 3). The conditions required in the definition of a dependence complex above imply that a node of κ has 3 'entering' edges iff it belongs to $col(\pi, C)$. Moreover, a node of this kind is twice a head node and once a tail node: this determines which of the 'branches' outgoing the node is a rope.

LEMMA 2.1 Let G be a DAG over N, $C \subseteq N$ and $a, b \in N \setminus C$ are distinct nodes. Then $a \not\perp\!\!\!\perp b \,|\, C\ [G]$ iff there exists a dependence complex in G between a and b for C.

Proof: We prove that $a \not\perp\!\!\!\perp b \,|\, C\ [G]$ implies the existence of the complex, the converse is trivial. Let us choose a path between a and b in G which is active w.r.t. C and has minimal number of colliders among all paths in G of this kind. Denote the chosen path $a = w_1, \ldots, w_k = b$, $k \geq 1$ by π and choose for every $d \in col(\pi, C)$ a directed path $\rho(d)$ from d to a node in C which has minimal number of arrows among all paths of this sort. This choice ensures that the chosen path $\rho(d) : t_1, \ldots, t_r$, $r \geq 2$ is outside C with exception of t_r.

We verify by contradiction that $\rho(d)$ does not share a node with π except $d = w_i$, $1 < i < k$. If this is not the case then choose minimal $2 \leq s \leq r$ for which $t_s \in \{w_1, \ldots, w_k\}$. One has $t_s = w_j$ for some $1 \leq j \leq k$, $j \neq i$. One can assume without loss of generality that $j > i$ as otherwise one can interchange a and b and replace π by the path $\tilde{w}_1, \ldots, \tilde{w}_k$ where $\tilde{w}_i = w_{k+1-i}$ for $i = 1, \ldots, k$. Let us introduce a new path σ in G which is made from π by replacement of its section $w_i, \ldots, w_j$ by respective section of $\rho(d)$, namely by $t_1, \ldots, t_s$. To get a contradiction one needs to verify that σ is active w.r.t. C and has lower number of colliders than π. As π and σ coincide outside the section between $w_j = t_1$ and $w_j = t_s$, the nodes $t_1, \ldots, t_{s-1}$ are non-colliders of σ outside C and t_s has a descendant t_r in C to verify that σ is active w.r.t. C it suffices to show that $w_j \notin C$ in case w_j is a non-collider of σ. For the same reasons and because w_i is a collider of π to verify the second claim it suffices to show that π has at least one collider among nodes $w_{i+1}, \ldots, w_j$ in case w_j is a collider of σ. To show these facts 3 subcases can be distinguished.

1. If $s = r$ then $w_j = t_r \in C$ and w_j is a collider of π (since π is active w.r.t. C). In particular, w_j is a collider of σ as well.

2. If $s < r$ and w_j is a non-collider of σ then $w_j = t_s \notin C$ (since $\rho(d)$ is outside C with exception of t_r).

3. If $s < r$ and w_j is a collider of σ then assume for contradiction that π has no collider among nodes $w_{i+1}, \ldots, w_j$. The fact $w_{j+1} \to w_j$ in G and this assumption allows to derive stepwise that $w_j, w_{j-1}, \ldots, w_i$ is a directed path in G. Since $w_i = t_1, \ldots, t_s = w_i$ is a directed path in G as well (as a part of $\rho(d)$) a directed cycle in G exists which contradicts the assumption.

In either case we have shown that σ is active path with lower number of colliders that π which contradicts the assumption about π. Thus, $\rho(d)$ does not share an node with π which means it is a rope for d with respect to π.

To conclude the proof it remains to be shown that two ropes for different colliders in $col(\pi, C)$ cannot share a node. To get a contradiction, suppose that a rope $\rho(d) : t_1, \ldots, t_r$, $r \geq 2$ for $d = w_i$ shares a node with a rope $\rho(e) : v_1, \ldots, v_l$, $l \geq 2$ for $e = w_j$, $i < j < k$ where $e \neq d$. Choose minimal $1 \leq s \leq r$ for which $t_s \in \{v_1 \ldots, v_l\}$. Since $\rho(e)$ is a rope with respect to π one has $2 \leq s$. Let $t_s = v_m$ for $1 \leq m \leq l$. Since $\rho(d)$ is a rope with respect to π one has $2 \leq m$. Let us introduce a new path φ in G which is made from π by replacement of its section $w_i, \ldots, w_j$ by the path $t_1, \ldots, t_s = v_m, \ldots, v_1$ in G. As $t_1, \ldots, t_{s-1}$ and $v_1, \ldots, v_{m-1}$ are non-colliders of φ outside C and $t_s = v_m$ is a collider of φ having a descendant t_r in C the path φ is active w.r.t. C. There is no other collider of φ between w_i and w_j (including them) while π has at least two colliders among $w_i, \ldots, w_j$. Therefore, φ has lower number of colliders than π which contradicts the assumption about π. Thus $\rho(d)$ and $\rho(e)$ do not share a node. □

REMARK 1 We believe that Lemma 2.1 can be generalized to the case of general directed graphs provided that the definition of d-separation is generalized properly. The idea of the proof is the same but to overcome possible difficulties one should weight every node by a minimal length of a path to C and to weight every path active w.r.t. C by the sum of weights of its colliders. The main modification is that one takes an active path of minimal weight. Note that the concept of dependence complex corresponds to the concept 'path-with-tails' mentioned by Matúš in [7].

CONSEQUENCE 2.1 Let G be a DAG over N and $\langle A, B|C\rangle \in \mathcal{T}(N)$. Then the following conditions are equivalent.

(i) $\langle A, B|C\rangle$ is not represented in G according to the d-separation criterion,

(ii) there exists a dependence complex for C in G between a node $a \in A$ and a node $b \in B$,

(iii) $\langle A, B|C\rangle$ is not represented in G according to the moralization criterion.

Proof: The equivalence (i)$\Leftrightarrow$(ii) is a direct consequence of Lemma 2.1 while the equivalence (i)$\Leftrightarrow$(iii) is shown in [6]. □

In particular, the notation $A \not\perp\!\!\!\perp B \,|\, [G]$ can be used to indicate validity of any of 3 conditions from Consequence 2.1.

2.3 Other preliminaries

The following specific notation for certain composite dependence statements will be useful in this report. Given a DAG G over N, distinct nodes $u, v \in N$ and disjoint sets $S, T \subseteq N \setminus \{u, v\}$ the symbol $u \not\perp\!\!\!\perp v \,|\, +T - S\ [G]$ will be interpreted as the condition

$$\forall W \text{ such that } T \subseteq W \subseteq N \setminus \{u, v\} \cup S \quad \text{one has} \quad u \not\perp\!\!\!\perp v \,|\, W\ [G]\,.$$

In words, u and v are (conditionally) dependent in G given any superset of T which is disjoint with S. In case that T respectively S is empty the symbols $+T$ respectively $-S$ are omitted; if both T and S is empty we write $\star$ instead of $+T - S$. In particular, the following two symbols will be sometimes used

$$u \not\perp\!\!\!\perp v \,|\, \star\ [G] \quad \equiv \quad \forall W \text{ such that } W \subseteq N \setminus \{u, v\} \quad u \not\perp\!\!\!\perp v \,|\, W\ [G]$$

for distinct nodes $u, v \in N$, and

$$u \not\perp\!\!\!\perp v \,|\, +w\ [G] \quad \equiv \quad \forall W \text{ such that } \{w\} \subseteq W \subseteq N \setminus \{u, v\} \quad u \not\perp\!\!\!\perp v \,|\, W\ [G]$$

for distinct nodes $u, v, w \in N$. We give a certain graphical characterization of composite dependence statements of this kind below. These auxiliary results were proved in [11] in the context of chain graphs but we recall their proofs in the special case of DAGs for reader's convenience.

LEMMA 2.2 Let G be a DAG over N and $u, v \in N$ are distinct nodes. Then

$$u \perp\!\!\!\perp v \,|\, pa_G(u)pa_G(v) \;\; [G] \quad \text{whenever} \;\; u \not\leftrightarrow v \;[G]\,. \tag{1}$$

Proof: We apply the moralization criterion to $\langle u, v|T\rangle$ where $T = pa_G(u)pa_G(v)$. Evidently $an_G(\{u,v\} \cup T) = an_G(\{u,v\}) \equiv D$ and one should consider the induced subgraph $K = G_D$. Let us verify by contradiction that either $ch_K(u) = \emptyset$ or $ch_K(v) = \emptyset$ is empty. Indeed, if $u \to t$ $[K]$ for a node t then owing to $t \in D$ a directed path from t to $\{u, v\}$ in G exists. Since G is acyclic the path leads to v which together with the fact $u \to t$ $[G]$ implies $u \in an_G(v)$. Similarly, $v \to s$ $[K]$ for a node s implies by $v \to s$ $[G]$ and acyclicity of G that there exists a directed path from s to u in G which means $v \in an_G(u)$ as $v \to s$ $[G]$. This contradicts the fact $u \in an_G(v)$ as G is acyclic.

Thus, one can assume without loss of generality that $ch_K(u) = \emptyset$. This implies that no immorality $(u, v) \rightsquigarrow w$ exists in K which means no new 'neighbours' of u are added when the moral graph $H = K^{mor}$ is constructed. Thus, u has $pa_G(u)$ as the set of its 'neighbours' in H and every path from u to v contains a node of $pa_G(u) \subseteq T$. $\square$

LEMMA 2.3 Let G be a DAG over N and $u, v \in N$ are distinct nodes. Then

$$u \leftrightarrow v \;[G] \;\; \text{iff} \;\; u \top\!\!\!\top v \,|\, \star \;\; [G]\,. \tag{2}$$

Proof: Suppose $u \leftrightarrow v$ $[G]$; the line $\{u,v\}$ occurs in the moral graph G_D where $D = an_G(\{u,v\} \cup W)$ for every $W \subseteq N \setminus \{u,v\}$. Then, $\langle u, v|W\rangle$ is not represented in G according to the moralization criterion. The converse implication follows from Lemma 2.2. $\square$

LEMMA 2.4 Let G be a DAG over N and $u, v, w \in N$ are distinct nodes such that $u \leftrightarrow w$ $[G]$, $v \leftrightarrow w$ $[G]$ and $u \not\leftrightarrow v$ $[G]$. Then

$$(u, v) \rightsquigarrow w \;[G] \;\; \text{iff} \;\; u \top\!\!\!\top v \,|\, +w \;[G]\,. \tag{3}$$

Proof: Suppose $(u, v) \rightsquigarrow w$ $[G]$; the line $\{u, v\}$ occurs in the moral graph of G_D where $D = an_G(\{u,v\} \cup W)$ for every $W \subseteq N \setminus \{u, v\}$ with $w \in W$. Hence, $u \top\!\!\!\top v \,|\, W$ $[G]$. To prove sufficiency suppose by contradiction that the induced subgraph on $\{u, v, w\}$ is not an immorality in G. The assumptions of Lemma 2.4 then imply $w \in pa_G(u)pa_G(v)$. Then, Lemma 2.2 leads to contradiction. $\square$

3 Equivalence problem

In this section we deal with a well understood special case of the inclusion problem - the equivalence problem. It is the problem how to recognize whether two given DAGs K and L over N induce the same independence model. It is of special importance to have an easy rule how to recognize that two DAGs are equivalent in this sense (the simplicity of a rule may differ when considering people and a computer to use it) and an easy way to get from L to K in terms of some elementary operations on graphs. These issues were treated in [15], [4] and [1]. Another very important aspect is the ability of generating all DAGs which are equivalent to a given DAG. To cope with the questions that arose, we need following definition.

DEFINITION 3.1 By a *legal arrow reversal* we understand the change of a DAG L into a directed graph K by replacement of an arrow $a \to b$ (in L) by $b \to a$ (in K) under the condition that $pa_L(a) \cup a = pa_L(b)$ (here $a, b \in N$ are some distinct nodes).

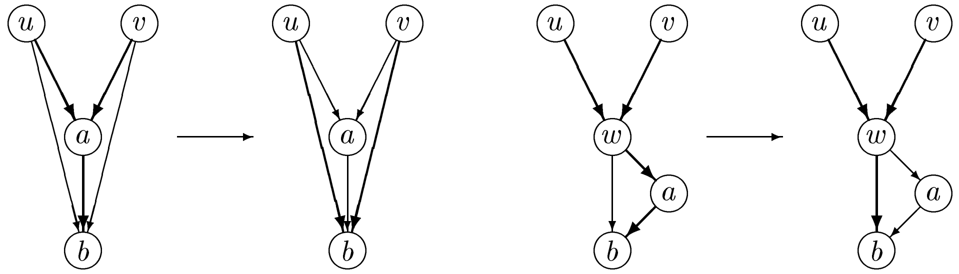

Figure 1: Rope modification (shortening).

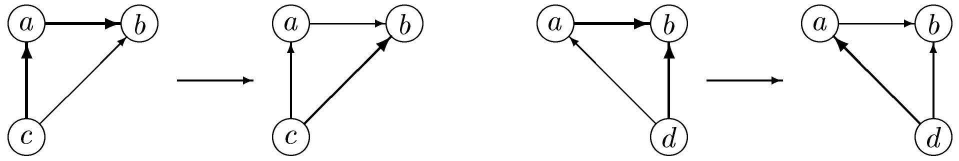

Figure 2: Path shortening.

Note that Chickering [1] and Meek [8] used another terminology: $a \to b$ is covered in L.

OBSERVATION 1 The result of legal arrow reversal is DAG.

Proof: Let L be the original DAG and K the result of legal arrow reversal applied to L. We prove the acyclicity of K by contradiction. Suppose that K has a directed cycle. The cycle necessarily contains the 'reversed' arrow $b \to a$. It also contains an arrow having b as a head node: $c \to b$, $c \neq a$. Hence $c \in pa_K(b)$ implies $c \in pa_L(b)$ and one gets $c \in pa_L(a)$ by Definition 3.1. Thus $c \to a$ in K which implies that there is a shorter cycle in K having $c \to a$ instead of $c \to b \to a$. But this cycle must be in L which contradicts the assumption. $\square$

LEMMA 3.1 Let K and L are DAGs over N such that K is obtained from L by legal arrow reversal. Then $\mathcal{I}(K) = \mathcal{I}(L)$.

Proof: We show that $\mathcal{D}(K) = \mathcal{D}(L)$. Since the role of K and L is interchangeable it suffices to verify $\mathcal{D}(L) \subseteq \mathcal{D}(K)$. Assume that the arrow $a \to b$ in L is changed into $a \leftarrow b$ in K. This means

$$pa_L(b) = pa_L(a) \cup \{a\}\,. \tag{4}$$

Suppose $A \perp\!\!\!\perp B \,|\, C\ [L]$, there exists C-complex in L between A and B. Without loss of generality, consider a C-complex κ which involves the minimal number of edges among complexes of this type. Then $a \to b$ is not an edge of any rope of κ, since otherwise (4) implies that κ can be modified to get a C-complex in L that has not $a \to b$ as an edge as shown in Figure 1. If $a \to b$ belongs to the active path π in κ then no arrow $c \to a$ and no arrow $b \leftarrow d$ is in π since otherwise π can be shorthened (and therefore κ modified) as shown in Figure 2. Note that in the latter case ($b \leftarrow d$) we utilize the fact that the former case ($c \to a$) is already excluded. These two facts imply that κ remains a C-complex in K after replacement of $a \to b$ in L by $a \leftarrow b$ in K; the argument is: if $a \to b$ is in π then neither a nor b is a collider of π both in L and K. Hence $A \perp\!\!\!\perp B \,|\, C\ [K]$. $\square$

Lemma 3.2 Supposing K and L are DAGs over N the following three conditions are equivalent

(1) $\mathcal{I}(K) = \mathcal{I}(L)$,

(2) $\mathcal{E}(K) = \mathcal{E}(L)$ and the graphs K and L have the same immoralities,

(3) there exists a sequence $G_1, \ldots, G_m$, $m \geq 1$ of DAGs over N such that $G_1 = L, G_m = K$ and G_{i+1} is obtained from G_i by legal arrow reversal for $i = 1, \ldots, m-1$.

Note that the equivalence (1) $\Leftrightarrow$ (2) was proved in [15], in more general framework of chain graphs in [4]; the equivalence (1) $\Leftrightarrow$ (3) was proved in [1] and [5].

Proof: We show (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1). The implication (1) $\Rightarrow$ (2) is an easy consequence of Lemmas 2.3 and 2.4 as $\mathcal{I}(K) = \mathcal{I}(L)$ is equivalent to $\mathcal{D}(K) = \mathcal{D}(L)$.

The proof of (2) $\Rightarrow$ (3) is done by induction on $|N|$. The induction hypothesis for $n \geq 1$ is that (2) $\Rightarrow$ (3) holds for any pair of graphs K, L over N with $|N| \leq n$. It is evident for $n = 1$. Assume $n = |N| \geq 2$ and that the implication holds for DAGs over N' with $|N'| < n$. The first step is to choose a terminal node $t \in N$ in K and put $P = pa_L(t), C = ch_L(t)$. Observe that $\mathcal{E}(K) = \mathcal{E}(L)$ implies $pa_K(t) = P \cup C$. One can distinguish two cases

I. $C = \emptyset$ which means $pa_L(t) = pa_K(t)$,

II. $C \neq \emptyset$ which means $pa_K(t) \setminus pa_L(t) \neq \emptyset$.

If $C = \emptyset$ then introduce L' respectively K' as the induced subgraph of L respectively K over $N' \equiv N \setminus \{t\}$. By the induction hypothesis, a desired sequence of $L' = G'_1, \ldots, G'_m = K'$, $m \geq 1$ exists. Introduce G_i as a graph over N obtained from G'_i by adding a bunch of arrows from nodes of P to t for $i = 1, \ldots, m$. It is easily verified that G_{i+1} is obtained from G_i by legal arrow reversal for $i = 1, \ldots, m-1$.

If $C \neq \emptyset$ then choose $c \in C$ such that no other $c' \in C$ is an ancestor of c in L. This choice is always possible and ensures that $pa_L(c) \cap C = \emptyset$. The second step is to observe $P \subseteq pa_L(c)$. Indeed, suppose that $p \not\leftrightarrow c$ $[L]$ for some $p \in P$. Then, $p \not\leftrightarrow c$ $[K]$, $p \leftrightarrow t$ $[K]$ and $c \leftrightarrow t$ $[K]$ by $\mathcal{E}(K) = \mathcal{E}(L)$. Since t is a terminal node in K one has $(p, c) \rightsquigarrow t$ $[K]$ and $(p, c) \rightsquigarrow t$ $[L]$ by (2). This however contradicts the fact $t \to c$ in L. Thus, necessarily $p \leftrightarrow c$ $[L]$. Since L is acyclic and $p \to t \to c$ in L it implies $p \to c$ in L. Another observation is that $pa_L(c) \subseteq P \cup \{t\}$. Indeed, suppose that there exists $y \in N \setminus P$, $y \neq t$ such that $y \to c$ in L (see Figure 3 for illustration where, however, arrows from P to C are omitted for sake of lucidity). Since $y \notin P$ and $y \notin C$ (because of the choice of c) one has $t \not\leftrightarrow y$ $[L]$. Thus $y \to c \leftarrow t$ in L implies $(y, t) \rightsquigarrow c$ $[L]$ and $(y, t) \rightsquigarrow c$ $[K]$ by (2). This contradict the fact $c \to t$ in K. Therefore, $pa_L(c) = P \cup \{t\}$ necessarily.

The fact $pa_L(c) = pa_L(t) \cup \{t\}$ means that the arrow $t \to c$ in L can be legally reversed. The procedure can be repeated until all arrows in C are legally reversed. Thus, a sequence $L = G_1, \ldots, G_k$ is constructed by legal arrow reversal such that t has the same parent set in G_k as K. Then, the case **I.** occurs for the pair (G_k, K) which was already treated. This concludes the induction step.

The proof of (3) $\Rightarrow$ (1) follows from repetitive application of Lemma 3.1. $\square$

Remark 2 Another method of computer testing of equivalence of DAGs over N is partially based on the approach from [12] where certain integer-valued functions on the power set of N called *structural imsets* over N are used to describe independence models over N. Every structural imset η over N induces a certain independence model $\mathcal{I}(\eta) \subseteq \mathcal{T}(N)$ by means of a

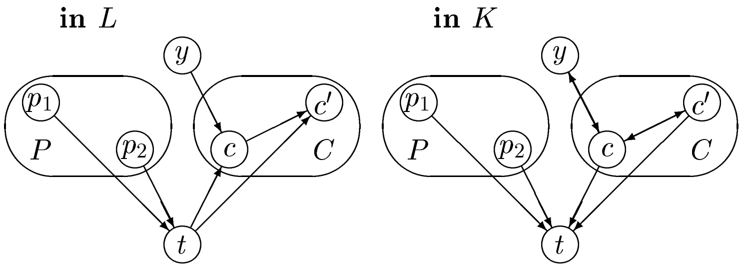

Figure 3: Proof of $pa_L(c) \subseteq P \cup \{t\}$ by contradiction.

certain algebraic criterion. The point is that every DAG G over N can be associated with a structural imset η over N in such a way that $\mathcal{I}(\eta) = \mathcal{I}(G)$. For example, the imset η_G defined by the following formula has this property:

$$\eta_G(S) \;=\; \delta(S,N) - \delta(S,\emptyset) + \sum_{i\in N} \{\, \delta(S, pa_G(i)) - \delta(S, pa_G(i)\cup\{i\})\,\} \quad \text{for } S \subseteq N\,, \qquad (5)$$

where we use the convention $\delta(A,B) = 1$ in case $A = B$ and $\delta(A,B) = 0$ in case $A \neq B$. Let us warn the reader that η_G is not the only structural imset η over N satisfying $\mathcal{I}(G) = \mathcal{I}(\eta)$. However, the claim above (which is proved in [12]; for rough idea of the proof see Example 3.5 in [10]) implies that two DAGs K and L are equivalent whenever $\eta_K = \eta_L$. The converse implication can be derived as a consequence of Lemma 3.2. Thus, K and L are equivalent iff $\eta_K = \eta_L$ which can be expressed in the form

$$\sum_{i\in N} \{\, \delta(S, pa_L(i)\cup\{i\}) - \delta(S, pa_L(i))\,\} \;=\; \sum_{i\in N} \{\, \delta(S, pa_K(i)\cup\{i\}) - \delta(S, pa_K(i))\,\}$$

for every $S \subseteq N$. Therefore, one can test equivalence of DAGs in the following manner. One writes a 'formal ratio' for every DAG G over N as follows: in the nominator one lists sets $pa_G(i) \cup \{i\}$ for $i \in N$ while in the denominator one lists the sets $pa_G(i)$ for $i \in N$. Then cancellation is performed: one occurrence of a set $A \subseteq N$ in the denominator is cancelled against one occurrence of A in the nominator. For example the DAG L in Figure 5 induces the following 'ratio':

$$\frac{a * ab * bc * cd}{\emptyset * a * b * c} = \frac{ab * bc * cd}{\emptyset * b * c}\,.$$

The DAGs over N are equivalent iff they lead to the same formal ratio after cancellation. For example, the graph which is obtained from the graph L in Figure 5 by reversal of all arrows has the following ratio

$$\frac{ab * bc * cd * d}{b * c * d * \emptyset} = \frac{ab * bc * cd}{\emptyset * b * c}$$

and therefore it is equivalent to L.

Consequence 3.1 Let K, L be DAGs over N such that $\mathcal{I}(K) = \mathcal{I}(L)$. Then $\eta_K = \eta_L$.

Proof: By Lemma 3.2 it suffices to show that $\eta_K = \eta_L$ whenever K is obtained from L by legal reversal of an arrow $a \to b$ in L. Observe that $pa_K(u) = pa_L(u)$ for any $u \in N \setminus \{a,b\}$ which means that for $\eta_K = \eta_L$ one needs to show

$$\sum_{i\in\{a,b\}} \{\, \delta(S, pa_L(i)\cup\{i\}) - \delta(S, pa_L(i))\,\} \;=\; \sum_{i\in\{a,b\}} \{\, \delta(S, pa_K(i)\cup\{i\}) - \delta(S, pa_K(i))\,\}$$

for any $S \subseteq N$. This is evident as $pa_L(a) = pa_K(b) = P$, $pa_L(b) = P \cup \{a\}$ and $pa_K(a) = P \cup \{b\}$ for a certain set $P \subseteq N$. $\square$

4 Inclusion problem

The inclusion problem can be formulated as follows. Given a set of variables N and two DAGs K and L over N is there an elegant graphical characterization of $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ which gives an efficient criterion to decide?

In the previous section we recalled the well-known graphical characterization of *equivalence* of DAGs K and L, that is characterization of $\mathcal{I}(K) = \mathcal{I}(L)$, by means of the underlying graph and immoralities. These conditions intuitively suggests that there exists a similar set of conditions to characterize inclusion.

Note that not every DAG is uniquely determined by its induced independence model. For example, the DAGs K and L over $N = \{a, b\}$, where K has only the arrow $a \to b$ and L has only the arrow $b \to a$ induce the same independece model. DAGs inducing the same model are said to be in the same *equivalence class*. Every equivalence class $\mathsf{K} = \{K_1, \ldots, K_n\}$, $n \geq 1$ is uniquely determined by the shared independence model $\mathcal{I}(\mathsf{K}) \equiv \mathcal{I}(K_i)$ for $i = 1, \ldots, n$. Thus, equivalence classes can be naturally ordered by inclusion of their induced models. In particular, the collection of all equivalence classes over a given set of variables N forms a poset (i.e., **p**artially **o**rdered **set**). It can be visualized by means of Hasse diagram. In that diagram hyper-nodes represent equivalence classes of DAGs and two classes K and L are connected by a link if K is included in L, which means $\mathcal{I}(\mathsf{K}) \subset \mathcal{I}(\mathsf{L})$, and no third class G is between them, which means $\mathcal{I}(\mathsf{K}) \subset \mathcal{I}(G) \subset \mathcal{I}(\mathsf{L})$ for none DAG G. Figure 4 shows Hasse diagram of this poset for three variables (= nodes).

It is clear that the maximum number DAGs in the equivalence classes grows exponentially with $|N|$. This is because the class of DAGs whose underlying graph is a complete graph over $|N|$ has $|N|!$ elements: each ordering of the variables results in a unique DAG from this class. The main problem is efficient testing of inclusion $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ for two DAGs K and L over N. Using the notion of input list (defined shortly) and its properties, we can formulate a polynomial time algorithm.

Let N be a set of variables, G be a DAG over N and $\theta : u_1, \ldots, u_n$, $n \geq 1$ a causal ordering for G on N. For every $u \in N$, $u = u_i$ for $1 \leq i \leq n$, the set of *predecessors* of u in θ is the set

$$pre_\theta(u) = \{v \in N\,;\, \theta(v) < \theta(u)\} = \{u_j;\, 1 \leq j < i\}\,.$$

An corresponding *input list* $\mathcal{L}_{G,\theta}$ is the list of independence statements

$$\mathcal{L}_{G,\theta} = \{\, \langle u, pre_\theta(u) \setminus pa_G(u) \,|\, pa_G(u) \rangle\,;\, u \in N\,\}\,.$$

Let A, B, C, D be sets. The so called *graphoid axioms* are the following set of derivation rules for elements of $\mathcal{T}(N)$

- triviality: $\langle A, \emptyset | C \rangle$
- symmetry: $\langle A, B | C \rangle \Rightarrow \langle B, A | C \rangle$
- decomposition: $\langle A, BD | C \rangle \Rightarrow \langle A, B | C \rangle$
- weak union: $\langle A, BD | C \rangle \Rightarrow \langle A, B | CD \rangle$

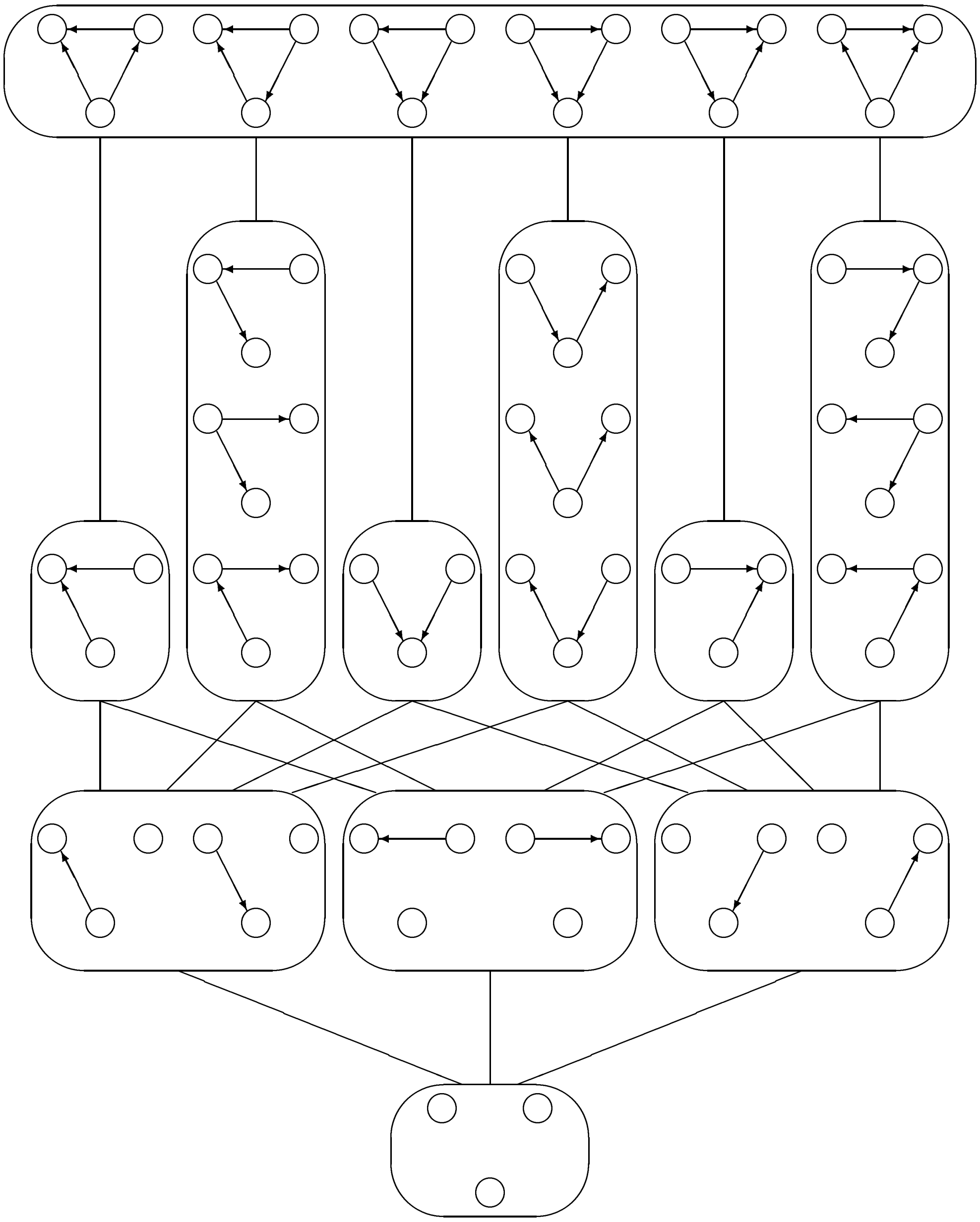

Figure 4: Hasse diagram of the poset of equivalence classes of DAGs with 3 nodes.

- contraction: $\langle A, B|CD\rangle$ and $\langle A, D|C\rangle \Rightarrow \langle A, BD|C\rangle$
- intersection: $\langle A, B|CD\rangle$ and $\langle A, D|BC\rangle \Rightarrow \langle A, BD|C\rangle$

For d-separation in DAGs, the graphoid axioms hold [9]. In other words $\mathcal{I}(G)$ is closed under graphoid axioms. So, if we have $A \perp\!\!\!\perp BD \,|\, C$ $[G]$ then by decomposition, we also have $A \perp\!\!\!\perp B \,|\, BD$ $[G]$. For d-separation in DAGs more rules are known. One of them is

- composition: $A \perp\!\!\!\perp B \,|\, C$ $[G]$ and $A \perp\!\!\!\perp D \,|\, C$ $[G] \Rightarrow A \perp\!\!\!\perp BD \,|\, C$ $[G]$.

Input lists have an interesting property that the closure of an input list $\mathcal{L}_{G,\theta}$ of a DAG G under the graphoid axioms is equal to the model $\mathcal{I}(G)$ of G [14]. This property suggests the following way for testing inclusion.

LEMMA 4.1 Let K and L be DAGs over a set of variables N. Let $\mathcal{L}_{K,\theta}$ be an input list for K and θ a causal ordering for K. Then $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ iff $\mathcal{L}_{K,\theta} \subseteq \mathcal{I}(L)$.

Proof: Observe that $\mathcal{I}(K)$ is equal to the closure of $\mathcal{L}_{K,\theta}$ under the graphoid axioms [14]. The graphoid axioms hold for d-separation in DAGs [9], so given that $\mathcal{L}_{K,\theta} \subseteq \mathcal{I}(L)$ we conclude $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. $\square$

The way to use this lemma in a computer implementation is to test for each disjoint triplet $\langle A, B|C\rangle \in \mathcal{L}_{K,\theta}$ whether $A \perp\!\!\!\perp B \,|\, C$ $[L]$. Testing a d-separation statement can be performed in $O(|N| + |\mathcal{E}(L)|)$ steps. Since $\mathcal{L}_{K,\theta}$ contains $|N|$ statements at most, the algorithm would be of time complexity $O(|N| \cdot (|N| + |\mathcal{E}(L)|))$. So, this gives an efficient polynomial complexity algorithm. Unfortunately, this characterization does not provide much insight in the structure of the DAG K with respect to L. Also, this result seems to be difficult to extend and to generalize as well. So, this characterization satisfies the condition required in the formulation of the inclusion problem that testing should be efficient, but it does not satisfy the requirement that it should be elegant. For this reason we do not consider the inclusion problem to be solved by this lemma.

5 Necessary conditions

In this section we formulate and compare various necessary conditions for validity $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ where K and L are DAGs over N.

5.1 Basic necessary conditions

Everybody who starts to deal with the inclusion problem gets almost immediately the following three *basic necessary conditions* for inclusion $\mathcal{I}(K) \subseteq \mathcal{I}(L)$:

(a) $u \leftrightarrow v$ $[L]$ $\Rightarrow$ $u \leftrightarrow v$ $[K]$,

($\tilde{\text{b}}$) $(u,v) \rightsquigarrow w$ $[L]$ $\Rightarrow$ $u \leftrightarrow v$ $[K]$ or $(u,v) \rightsquigarrow w$ $[K]$,

($\tilde{\text{c}}$) $(u,v) \rightsquigarrow w$ $[K]$ $\Rightarrow$ $u \not\leftrightarrow w$ $[L]$ or $w \not\leftrightarrow v$ $[L]$ or $(u,v) \rightsquigarrow w$ $[L]$.

Note that under assumption that (a) is valid the condition ($\tilde{\text{b}}$) can be formulated in the following way which may appear to be more suitable in some cases

(b) $u \rightarrow w \leftarrow v$ $[L]$ $\Rightarrow$ $u \leftrightarrow v$ $[K]$ or $u \rightarrow w \leftarrow v$ $[K]$.

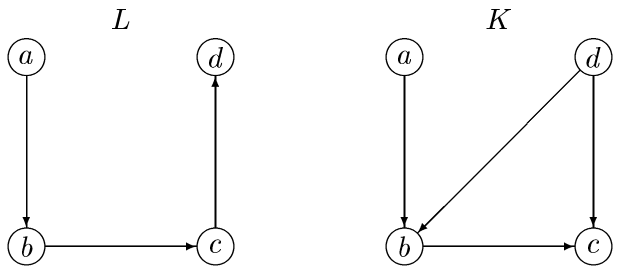

Figure 5: Basic necessary conditions are not sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$: the first example.

The necessity of these conditions follows immediately from Lemma 5.2 in Section 5.3 - see Consequence 5.1. The conditions are also sufficient in the following rather special case.

LEMMA 5.1 Suppose that K, L are DAGs over N such that $|\mathcal{E}(K)| \leq |\mathcal{E}(L)|$. Then the conditions (a), ($\tilde{\text{b}}$) and ($\tilde{\text{c}}$) are necessary and sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$.

Proof: Sufficiency of these conditions follows directly from Lemma 3.2: the condition (a) says $\mathcal{E}(L) \subseteq \mathcal{E}(K)$ which together with $|\mathcal{E}(K)| \leq |\mathcal{E}(L)|$ implies $\mathcal{E}(K) = \mathcal{E}(L)$. The conditions ($\tilde{\text{b}}$) and ($\tilde{\text{c}}$) then imply that K and L have the same immoralities. □

However these basic conditions are not sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ in general as an example in Figure 5 shows. Indeed, the conditions (a), ($\tilde{\text{b}}$) and ($\tilde{\text{c}}$) are evidently fulfilled in that case but one has $a \perp\!\!\!\perp d \,|\, \emptyset \; [K]$ while $a \top\!\!\!\top d \,|\, \emptyset \; [L]$ which implies $\neg\{\mathcal{I}(K) \subseteq \mathcal{I}(L)\}$.

5.2 Verma's conditions

Verma and Pearl formulated in one of their technical reports [13] (it was probably a preparatory text only) three conditions on DAGs K and L over N which they regarded as necessary and sufficient conditions for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$; but this is not the truth. Their conditions required

(i) $u \leftrightarrow v \; [L] \;\Rightarrow\; u \leftrightarrow v \; [K]$,

(ii) $u \top\!\!\!\top v \,|\, +w \; [L] \;\Rightarrow\; u \top\!\!\!\top v \,|\, +w \; [K]$,

($\widetilde{\text{iii}}$) $u \top\!\!\!\top v \,|\, +w \; [K],\; u \leftrightarrow w \; [L]\,,\; w \leftrightarrow v \; [L] \;\Rightarrow\; u \rightarrow w \leftarrow v \; [L]$ or $u \leftrightarrow v \; [K]$.

for distinct nodes $u, v, w \in N$. The meaning of the symbol which was in [13] used instead of $u \top\!\!\!\top v \,|\, +w \; [G]$ (for a DAG G over N) was described there by the phrase "u and v are common parents of some ancestor of w, or u and v are adjacent in G". Verma and Pearl showed that this graphical condition is equivalent to the requirement $u \top\!\!\!\top v \,|\, +w \; [G]$ introduced in Section 2.3; this is the truth - see Observation 6 in Section 5.4.2.

The condition ($\widetilde{\text{iii}}$) evidently implies the condition

(iii) $u \rightarrow w \leftarrow v \; [K],\; u \leftrightarrow w \; [L]\,,\; w \leftrightarrow v \; [L] \;\Rightarrow\; u \rightarrow w \leftarrow v \; [L]$ or $u \leftrightarrow v \; [K]$.

The necessity of conditions (i), (ii) and (iii) for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ follows from Lemma 5.2, Consequence 5.1 and Observation 6. Indeed, (i) is nothing but (a), (ii) is evidently necessary if the symbol $u \top\!\!\!\top v \,|\, +w$ is interpreted as a composite dependence statement and (iii) is equivalent to ($\tilde{\text{c}}$) under assumption that (a) holds. However as the example from Figure 5 shows they are not sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ in general. Another counter example is given in Figure 10.

5.3 Inclusion conditions

The most elegant formulation of our necessary conditions is as follows.

Definition 5.1 Let K and L are DAGs over N. We will call the following 3 conditions the *inclusion conditions for K in L* (here, u, v, w are distinct elements of N):

(a) $u \leftrightarrow v\ [L] \Rightarrow u \leftrightarrow v\ [K]$,

(b) $u \to w \leftarrow v\ [L] \Rightarrow u \leftrightarrow v\ [K]$ or $u \to w \leftarrow v\ [K]$,

($*$) $(u,v) \rightsquigarrow w\ [K] \Rightarrow u \perp\!\!\!\perp v \,|\, pa_K(u)pa_K(v)\ [L]$.

Lemma 5.2 Suppose that K, L are DAGs over N such that $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. Then the conditions (a), (b) and ($*$) from Definition 5.1 hold.

Proof: The condition $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ is equivalent to the condition $\mathcal{D}(L) \subseteq \mathcal{D}(K)$. It clearly implies (a) by Lemma 2.3. The condition (b) is better to verify in this form:

($\tilde{\text{b}}$) $(u,v) \rightsquigarrow w\ [L] \Rightarrow u \leftrightarrow v\ [K]$ or $(u,v) \rightsquigarrow w\ [K]$.

To evidence it one can apply Lemma 2.4 to $G = L$ then use $\mathcal{D}(L) \subseteq \mathcal{D}(K)$ and after that apply Lemma 2.4 to $G = K$ as one is sure that $u \leftrightarrow w \leftrightarrow v$ in K by (a). Finally, to evidence ($*$) observe that $(u,v) \rightsquigarrow w\ [K]$ implies $u \perp\!\!\!\perp v \,|\, pa_K(u)pa_K(v)\ [K]$ by Lemma 2.2 and hence $u \perp\!\!\!\perp v \,|\, pa_K(u)pa_K(v)\ [L]$ by $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. □

Consequence 5.1 Suppose that K, L are DAGs over N satisfying the inclusion conditions (a), (b) and ($*$). Then the basic necessary conditions from Section 5.1 hold.

Proof: It suffices to verify ($\tilde{\text{c}}$). Assume $(u,v) \rightsquigarrow w\ [K]$, put $W \equiv pa_K(u)pa_K(v)$ and observe $w \notin W$. Suppose for contradiction that the conclusion of ($\tilde{\text{c}}$) is not valid. That means either $u \leftrightarrow v\ [L]$ which contradicts the fact $u \not\leftrightarrow v\ [K]$ by (a) or there exists a path u, w, v in L without collider nodes. This path is then active w.r.t. W (as $w \notin W$) which means $u \not\perp\!\!\!\perp v \,|\, W\ [L]$. However, the condition ($*$) implies $u \perp\!\!\!\perp v \,|\, W\ [L]$ which contradicts that fact. Thus, the conclusion of ($\tilde{\text{c}}$) must hold. □

So, the inclusion conditions for K in L are necessary for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. However, we conjecture that they are also sufficient (see Conjecture 2). If this conjecture is correct this would solve the inclusion problem in the sense that it characterizes inclusion in an efficient manner and in a way that is as elegant as it can be expected knowing that any characterization must have a non-local aspect when 'going from K to L' (see Section 5.5).

5.4 Graphical necessary conditions

Unlike the condition ($*$) from Section 5.3, the basic necessary conditions from Section 5.1 represent nice graphical conditions. They are 'relatively local' in sense that their verification confines only in those nodes which are involved in their premises. Moreover, they are formulated in terms of invariants of equivalence classes. The aim of this section is to introduce a whole collection of such (relatively local) graphical conditions which (hopefully) together may occur to be equivalent to the inclusion conditions from Definition 5.1. The conditions are formulated 'in the direction from L to K' in the sense that their premises concern the graph L while their conclusions concern the graph K. Our analysis reveals three types conditions of this kind; but further types of these conditions can be expected - see Conjecture 3.

5.4.1 Open path conditions

DEFINITION 5.2 Let G be a DAG over N. A path π in G is called *open* if there is no collider node of π. We will write $w_1 - w_2 - \ldots - w_k$ $[G]$, $k \geq 1$ to denote an open path in G.

The reader can verify easily the following two facts.

OBSERVATION 2 Every section of an open path in G is an open path in G.

OBSERVATION 3 Suppose that π is an open path in G between distinct nodes $u, v \in N$. Then all nodes of π belong to $an_G(\{u,v\})$. If S is the set of internal nodes of π then $u \not\perp\!\!\!\perp v \,|\, -S$ $[G]$.

DEFINITION 5.3 Let K, L be DAGs over N and $\pi : w_1, \ldots, w_n$, $n \geq 1$ a path in L. By *shortening of π in K* is understood a path $\sigma : t_1, \ldots, t_r$, $1 \leq r \leq n$ in K such that there exists an increasing sequence of indices $1 = i_1 < i_2 < \ldots < i_r = n$ that $t_j = w_{i_j}$ for $j = 1, \ldots, r$. Note that in case $K = L = G$ the above defined concept reduces to the usual concept of shortening of a path in G. A path in G is then called *minimal* if it has no proper shortening in G (i.e., no other shortening except itself).

OBSERVATION 4 Let π be an open path in a DAG G over N. Then every its shortening in G is an open path as well.

Proof: Assume that $\pi : w_1, \ldots, w_n$, $n \geq 1$ is an open path in G and $\sigma : t_1, \ldots, t_r$, $1 \leq r \leq n$ is its shortening in G where $1 = i_1 < \ldots < i_r = n$ and $t_j = w_{i_j}$ for $j = 1, \ldots, r$. This can be shown by contradiction. Suppose that $t_{j-1} \to t_j \leftarrow t_{j+1}$ for some $1 < j < r$. Let $1 \leq k < i < l \leq n$ are indices for which $w_k = t_{j-1}$, $w_i = t_j$, $w_l = t_{j+1}$. Observe that $w_i \leftarrow w_{i+1}$ in G as otherwise $w_i \to w_{i+1}$ together with the assumption of absence of collider nodes in the section $w_i, \ldots, w_l$ implies $w_i \to \ldots \to w_l$ in G which together with $w_l = t_{j+1} \to t_j = w_i$ contradicts acyclity of G. A similar consideration leads to the conclusion $w_{i-1} \to w_i$ in G. Therefore, w_i is a collider node in π which contradicts the assumption. $\square$

DEFINITION 5.4 Let K and L are DAGs over N. By the *open path condition for K in L* is understood the following requirement:

(A) Every open path in L has an open shortening in K.

LEMMA 5.3 Suppose that K, L are DAGs over N satisfying the inclusion conditions (a) and $(*)$. Then the condition (A) from Definition 5.4 holds.

Proof: Let $\pi : w_1, \ldots, w_k$, $k \geq 1$ be an open path in L. For every pair of distinct nodes $u, v \in N$ of π denote by $con_\pi(u,v)$ the set of nodes of π strictly between u and v, that is

$$con_\pi(u,v) = \{\, w_l \,;\, i < l < j \;\text{ where }\; u = w_i \;\text{ and }\; v = w_j \,\}\,.$$

The idea of the proof is to construct a sequence $\pi_0, \ldots, \pi_n$, $n \geq 0$ of paths such that for every $i = 0, \ldots, n$ the following holds:

1) π_i is a path in K,

2) π_i is a shortening of π (which is in L) and $\pi_0, \ldots, \pi_{i-1}$ (which are in K),

3) if $u \leftrightarrow v$ in π_i for $u, v \in N$ then $con_\pi(u,v) \subseteq ds_K(u) \cap ds_K(v)$,

and moreover

4) π_n is an open path (in K).

Clearly, π_n is the desired open shortening of π in K then. The first step is to observe that $\pi_0 = \pi$ satisfies the requirements 1)-3) for $i = 0$. Indeed, it follows from the condition (a). The essential step is to show (for every $i \geq 0$) that whenever π_i is a path satisfying 1)-3) then either π_i is an open path in K or there exists a proper shortening π_{i+1} of π_i satisfying 1)-3). Indeed, as every path has finitely many shortenings repetitive application of this step leads to the desired conclusion.

Thus, suppose that π_i satisfies 1)-3) and it is not an open path. Then π_i has a collider node which means $u \to w \leftarrow v$ in K for some distinct nodes $u, v, w \in N$. The first observation is

$$con_\pi(u,v) \subseteq ds_K(u) \cap ds_K(v). \tag{6}$$

Indeed, by 2) π_i is a shortening of π and therefore $con_\pi(u,v) = con_\pi(u,w) \cup \{w\} \cup con_\pi(w,v)$. By 3) and the fact $v \to w$ in K get $con_\pi(u,w) \subseteq ds_K(u) \cap ds_K(w) \subseteq ds_K(u) \cap ds_K(v)$. A similar consideration is valid for $con_\pi(w,v)$ and $w \in ch_K(u) \cap ch_K(v) \subseteq ds_K(u) \cap ds_K(v)$ then concludes the proof of (6). The second observation is $u \leftrightarrow v$ $[K]$. To this end suppose for contradiction the converse and derive $(u,v) \rightsquigarrow w$ $[K]$. Let $W = pa_K(u) pa_K(v)$ and observe $u \perp\!\!\!\perp v \,|\, W$ $[L]$ by $(*)$. On the other hand, (6) and the fact that K is acyclic implies $con_\pi(u,v) \cap W = \emptyset$. As $\{u,v\} \cap W = \emptyset$ in this case the section π' of π between u and v is outside W. It is an open path in L (by Observation 2) which means that it is active w.r.t. W and therefore $u \not\perp\!\!\!\perp v \,|\, W$ $[L]$. This contradicts the above mentioned fact. Therefore, necessarily $u \leftrightarrow v$ $[K]$. One can introduce π_{i+1} as a shortening of π_i: the section $u \to w \leftarrow v$ of π_i is replaced by the edge $u \leftrightarrow v$ of π_{i+1}. Validity of 1) and 2) for π_{i+1} is evident, 3) for π_{i+1} follows from 3) for π_i and (6). Thus, the above mentioned essential step was made and this concludes the proof of Lemma 5.3. $\square$

To illustrate the open path condition let us formulate some specific instances of the rule (A). They can be classified according to the length of the path in the premise of (A). The first three instances are as follows (u, v, w, t are distinct elements of N):

(A:1) $u - v$ $[L]$ $\Rightarrow$ $u - v$ $[K]$,

(A:2) $u - w - v$ $[L]$ $\Rightarrow$ $u - v$ $[K]$ or $u - w - v$ $[K]$,

(A:3) $u - w - t - v$ $[L]$ $\Rightarrow$ $u - v$ $[K]$ or $u - w - v$ $[K]$ or $u - t - v$ $[K]$ or $u - w - t - v$ $[K]$.

These three conditions are illustrated by Figure 6. Observe that (A:1) is nothing but (a) and the condition (A:2) is under (a) equivalent to the condition ($\tilde{c}$) from Section 5.1; the argument is that an implication $X \Rightarrow Y$ can be expressed in the form $\neg Y \Rightarrow \neg X$. The condition (A:3) makes it possible to evidence immediately that the pair of DAGs from Figure 5 does not satisfy the inclusion conditions. Indeed, $a \to b \to c \to d$ is an open path in L which has no open shortening in K.

Remark 3 Clearly, every condition (A:i) for $i = 1, 2, \ldots$ is 'relatively local'. Moreover, owing to Observation 4 the condition (A) can be formulated equivalently as follows.

(A*) Every minimal open path in L has a shortening in K which is a minimal open path (of course in K).

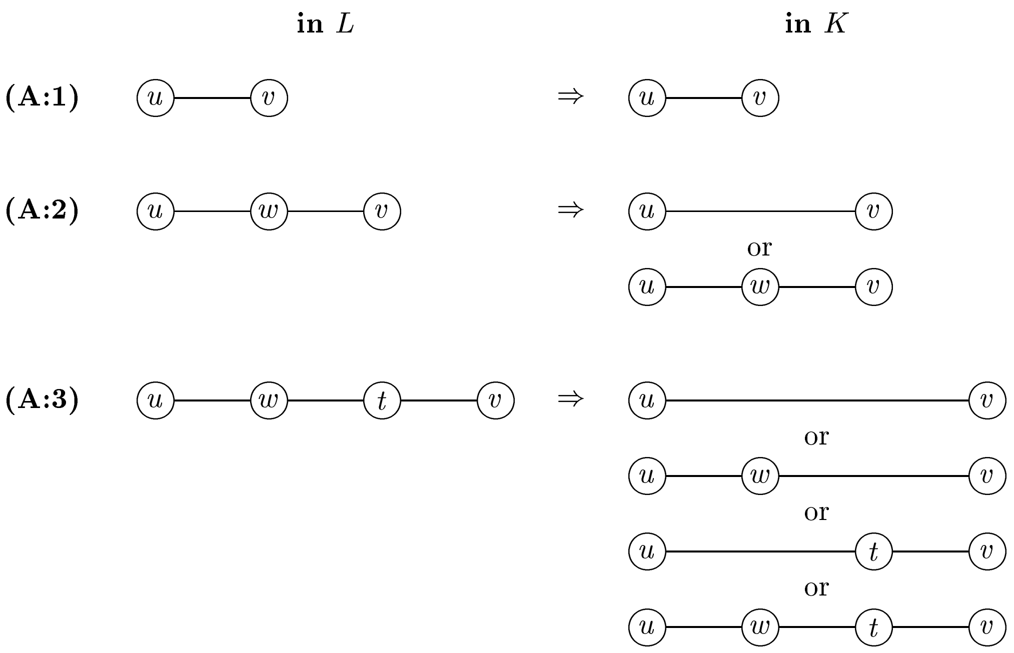

Figure 6: First three instances of the rule (A).

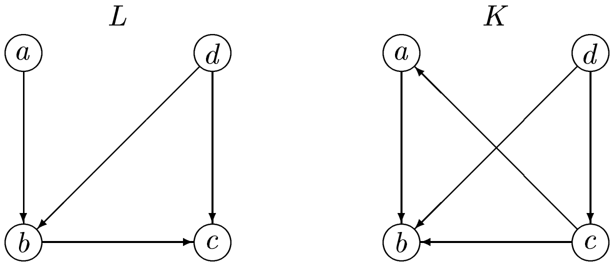

Figure 7: Basic necessary conditions are not sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$: the second example.

The point is that minimal open paths are invariants of Markov equivalence classes (unlike open paths) which means that all instances of (A*) are formulated in terms of invariants of equivalence classes.

Nevertheless, the condition (A) together with the condition (b) from Definition 5.1 are not strong enough to imply the inclusion conditions as the example in Figure 7 shows. Indeed, the conditions (A) and (b) are fulfilled in this case but one has $a \perp\!\!\!\perp d \,|\, c \; [K]$ while $a \not\perp\!\!\!\perp d \,|\, c \; [L]$ which implies $\neg\{\mathcal{I}(K) \subseteq \mathcal{I}(L)\}$. The condition $(*)$ is not satisfied since $(a,d) \rightsquigarrow b \; [K] \not\Rightarrow a \perp\!\!\!\perp d \,|\, c \; [L]$.

5.4.2 Dependence configuration conditions

Definition 5.5 Let G be a DAG over N and u, v, w are distinct nodes of G. By *dependence configuration* between u and v relative to w in G of order $n \geq 1$ is understood a certain subgraph λ of G. Its collection of edges consists of the edges of a simple collider path $u \to w_1 \leftarrow v$ and of the edges of a directed path $w_1 \to \ldots \to w_n = w$ which may be empty in case $n = 1$ (necessarily $w_i \notin \{u, v\}$ for $i = 1, \ldots, n$ by acyclicity of G). The set of nodes of λ is $\{u, v, w_1, \ldots, w_n\}$. The symbol $(u,v) \mapsto w_1,\ldots,w_n$ $[G]$ will be used to denote that G has a configuration of this form.

The reader can evidence the following facts.

OBSERVATION 5 If $(u,v) \mapsto w_1,\ldots,w_n$ $[G]$ for $n \geq 1$ then $w_n \in ds_G(u) \cap ds_G(v)$.

OBSERVATION 6 Let u,v,w be distinct nodes of G, the condition

$$u \not\perp\!\!\!\perp v \,|\, +w \; [G] \quad \equiv \quad \forall W \quad w \in W \subseteq N \setminus \{u,v\} \quad u \not\perp\!\!\!\perp v \,|\, W \; [G] \tag{7}$$

is equivalent to the requirement

$$u \leftrightarrow v \; [G] \text{ or } (u,v) \mapsto w_1,\ldots,w_n \; [G] \text{ for some sequence } w_1,\ldots,w_n = w. \tag{8}$$

Proof: The implication (8) $\Rightarrow$ (7) is straightforward. For converse implication suppose (7), $u \not\leftrightarrow v$ $[G]$ and introduce the set of *immoral connectors* between u and v as follows:

$$im_G(u,v) = \{w \in N;\, \exists\, t \in N \quad (u,v) \rightsquigarrow t \; [G] \text{ and } w \in ds_G(t)\}\,.$$

Let $T = N \setminus im_G(u,v) \cup \{u,v\}$. Since $N \setminus im_G(u,v) = an_G(N \setminus im_G(u,v))$ one can use the moralization criterion to show that $u \perp\!\!\!\perp v \,|\, T$ $[G]$. The condition (7) then implies $w \notin T$ which means $w \in im_G(u,v)$. Hence $(u,v) \mapsto w_1,\ldots,w_n$ for some $w_1,\ldots,w_n = w$, $n \geq 1$. □

DEFINITION 5.6 Let K,L be DAGs over N and $\lambda : (u,v) \mapsto w_1,\ldots,w_n$, $n \geq 1$ be a dependence configuration of order n in L. By a *subconfiguration of* λ *in* K is understood either the edge $u \leftrightarrow v$ $[K]$ or an configuration $\kappa : (u,v) \mapsto t_1,\ldots,t_r$, $1 \leq r \leq n$ in K such that there exists an increasing sequence of indices $1 \leq i_1 < i_2 \ldots < i_r = n$ that $t_j = w_{i_j}$ for $j = 1,\ldots,r$.
The definition above applies in case $K = L = G$ as well. Then, an edge $u \leftrightarrow v$ in G (more precisely the induced subgraph of G for $\{u,v\}$) can be regarded as a specific *total dependence configuration* between u and v relative w in G, namely the configuration of order 0. By a *generalized immorality* in G is understood a dependence configuration of order $n \geq 1$ which has no proper subconfiguration in G (i.e. no other subconfiguration except itself). It will be denoted by the symbol $(u,v) \rightsquigarrow w_1,\ldots,w_k$ $[G]$.

Thus, every subconfiguration of $(u,v) \mapsto w_1,\ldots,w_n$, $n \geq 1$ is determined by a subset A of $\{u,v,w_1,\ldots,w_n\}$, namely the set of its nodes. But only the set $A = \{u,v\}$ and the sets A satisfying $\{u,v,w_k\} \subseteq A \subseteq \{u,v,w_1,\ldots,w_k\}$ determine subconfigurations. Let us emphasize that the 'order' of nodes is a subconfiguration must follow the order in the given configuration. In other words, subconfiguration of a configuration λ is determined by the set of its nodes and by λ. For example, the configuration $\lambda : (a,b) \mapsto c,d,e$ in the DAG L from Figure 8 has in K subconfigurations $a \to b$ and $(a,b) \mapsto d,e$ only. The configuration $(a,b) \mapsto d,c,e$ in K is not supposed to be a subconfiguration of λ (which is in L). However, this strange phenomenon cannot occur if one considers subconfigurations in the same graph (because of acyclicity).

DEFINITION 5.7 Let K and L be DAGs over N. By a *dependence configuration condition* for K in L is understood the following claim:

(B) Every dependence configuration in L has a subconfiguration in K.

LEMMA 5.4 Suppose that K,L are DAGs over N satisfying the inclusion conditions (a),(b) and $(*)$ for K in L. Then condition (B) from Definition 5.7 holds.

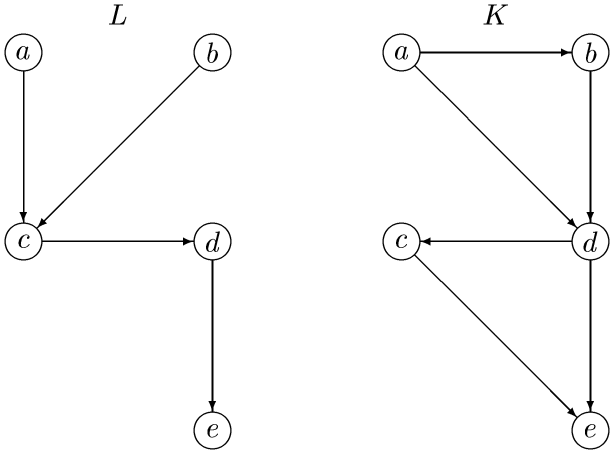

Figure 8: Configuration $(a, b) \mapsto d, c, e$ $[K]$ is not a subconfiguration of $(a, b) \mapsto c, d, e$ $[L]$.

Proof: The case of total configuration is covered by (a). Let $\lambda : (u, v) \mapsto w_1, \ldots, w_n$, $n \geq 1$ be a dependence configuration in L. We prove the existence of a subconfiguration in K by induction on n. If $n = 1$ then it follows from the condition (b). To verify the induction step for $n \geq 2$ assume that every configuration $(u, v) \mapsto t_1, \ldots, t_r$, $1 \leq r < n$ in L has a subconfiguration in K. According to Lemma 5.3 one is sure that the condition (A) holds. Let us distinguish the following four cases.

I. $u \leftrightarrow v$ $[K]$,

II. $u \not\leftrightarrow v$ $[K]$, $u \leftrightarrow w_n$ $[K]$ and $v \leftrightarrow w_n$ $[K]$,

III. $u \not\leftrightarrow v$ $[K]$ and $u \not\leftrightarrow w_n$ $[K]$,

IV. $u \not\leftrightarrow v$ $[K]$ and $v \not\leftrightarrow w_n$ $[K]$.

These cases are treated as follows.

I. If $u \leftrightarrow v$ $[K]$ then the conclusion is trivially valid.

II. If $u \not\leftrightarrow v$ $[K]$, $u \leftrightarrow w_n$ $[K]$ and $v \leftrightarrow w_n$ $[K]$ then observe $u \to w_1 \leftarrow v$ in L and by (b) derive $u \to w_1 \leftarrow v$ $[K]$. This implies $(u, v) \rightsquigarrow w_1$ $[K]$. Let us put $W = pa_K(u) pa_K(v)$ and by $(*)$ derive $u \perp\!\!\!\perp v \,|\, W$ $[L]$. On the other hand, by Observation 6 $(u, v) \mapsto w_1, \ldots, w_n$ $[L]$ implies that $u \not\perp\!\!\!\perp v \,|\, T$ $[L]$ for any $T \subseteq N \setminus \{u, v\}$ containing w_n. In particular, $w_n \notin W$. This together with $u \leftrightarrow w_n \leftrightarrow v$ $[K]$ implies $u \to w_n \leftarrow v$ in K and this is a subconfiguration of λ in K.

III. If $u \not\leftrightarrow v$ $[K]$ and $u \not\leftrightarrow w_n$ $[K]$ then observe that $\pi : u \to w_1 \to \ldots \to w_n$ is an open path in L. Using the open path condition (A) (see Lemma 5.3) and the fact $u \not\leftrightarrow w_n$ $[K]$ conclude that there exists $1 \leq l < n$ and a sequence of indices $l = l_1 < l_2 < \ldots < l_s = n$, $1 \leq s \leq n$ such that $\rho : u - w_l = w_{l_1} - w_{l_2} - \ldots - w_{l_s} = w_n$ is an open shortening of π in K. Consider the configuration $\kappa : (u, v) \mapsto w_1, \ldots, w_l$ in L. Since $l < n$ by the induction hypothesis observe that κ has a subconfiguration in K. It cannot be the total subconfiguration as $u \not\leftrightarrow v$ $[K]$ and therefore $w_l \in ds_K(u)$ by Observation 5 where $G = K$. Since $u \leftrightarrow w_l$ $[K]$ the acyclicity of K implies $u \to w_l$ in K. The fact that ρ is an open path in K then allows to show stepwise that it is a directed path in K. The above mentioned

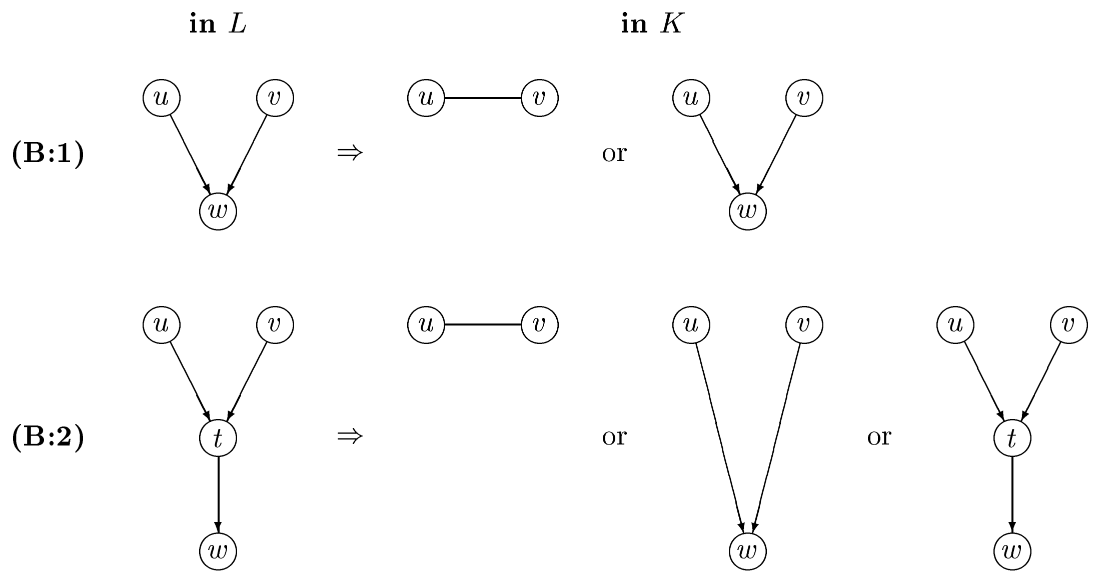

Figure 9: First two actual instances of the rule (B).

subconfiguration of κ in K then forms together with the section $w_{l_1} \to \ldots \to w_{l_s}$ of ρ a subconfiguration of λ in K.

IV. If $u \not\leftrightarrow v$ $[K]$ and $v \not\leftrightarrow w_n$ $[K]$ then use the procedure from **III.** where u is replaced by v.

Thus, the desired conclusion was achieved in all four possible cases. $\square$

To illustrate the dependence configuration condition let us formulate some specific instances of the rule (B). They are classified by the order of the dependence configuration in the premise of (B). The first three instances are as follows (u, v, w, t are distinct elements of N).

(B:0) $u \leftrightarrow v$ $[L]$ $\Rightarrow$ $u \leftrightarrow v$ $[K]$,

(B:1) $(u, v) \mapsto w$ $[L]$ $\Rightarrow$ $u \leftrightarrow v$ $[K]$ or $(u, v) \mapsto w$ $[K]$,

(B:2) $(u, v) \mapsto t, w$ $[L]$ $\Rightarrow$ $u \leftrightarrow v$ $[K]$ or $(u, v) \mapsto w$ $[K]$ or $(u, v) \mapsto t, w$ $[K]$.

The last two conditions are illustrated by Figure 9. Observe that (B:0) is nothing but (a) and (B:1) is nothing but (b). The condition (B:2) makes it possible to evidence immediately that the pair of DAGs from Figure 7 does not satisfy the inclusion conditions. Indeed, $(a, d) \mapsto b, c$ is a dependence configuration in L (actually, it is a generalized immorality in L) which has not a subconfiguration in K.

Remark 4 All conditions (B:i) for $i = 1, 2, \ldots$ are 'relatively local'. Since the relation 'being a subconfiguration' is transitive one can formulate the condition (B) equivalently as follows.

(B*) Every generalized immorality in L has a subconfiguration in K which is a generalized immorality in K.

The point is that generalized immoralities are invariants of Markov equivalence classes (unlike dependence configurations) which means that all instances of (B*) are formulated in terms of invariants of equivalence classes of DAGs.

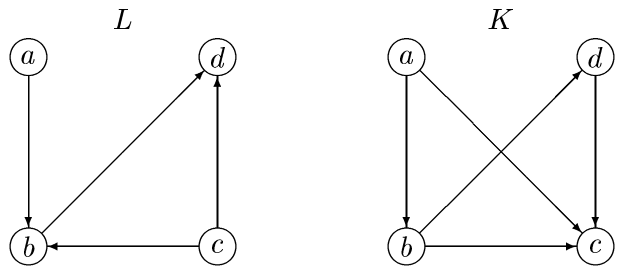

Figure 10: Basic necessary conditions are not sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$: the third example.

Unfortunately, the conditions (A) and (B) together are not strong enough to imply the inclusion conditions as shown by the example from Figure 10. Indeed, conditions (A) and (B) are fulfilled but one has $a \perp\!\!\!\perp d \,|\, b\; [K]$ while $a \top\!\!\!\top d \,|\, b\; [L]$ which implies $\neg\{\mathcal{I}(K) \subseteq \mathcal{I}(L)\}$. Condition $(*)$ is not valid since $(a,d) \rightsquigarrow c\; [K] \not\Rightarrow a \perp\!\!\!\perp d \,|\, b\; [L]$.

5.4.3 Mixed case

The preceding example indicates that there are other necessary conditions which combine open path conditions and dependence configuration conditions. One of such conditions is given below. First, we introduce the following notation for a DAG G over N and distinct nodes u, v, t, w.

$$u \to w \leftarrow t \leftrightarrow v\; [G] \quad \text{means} \quad \{\; u \to w \leftarrow t\; [G] \;\text{ and }\; t \leftrightarrow v\; [G]\; \}\,.$$

DEFINITION 5.8 Let K and L are DAGs over N. The simplest instance of a *mixed case condition* is the following requirement

(C:1) $u \to w \leftarrow t \leftrightarrow v\; [L] \;\;\Rightarrow\;\; u \leftrightarrow v\; [K]$ or $u - t - v\; [K]$ or $u \to w \leftarrow v\; [K]$ or $u \to w \leftarrow t \leftrightarrow v\; [K]$.

LEMMA 5.5 Suppose that K, L are DAGs over N satisfying the inclusion conditions for K in L. Then the condition (C:1) from Definition 5.8 holds.

Proof: By Lemmas 5.3 and 5.4 the conditions (A) and (B) holds. Suppose that $u \to w \leftarrow t\; [L]$ and $t \leftrightarrow v\; [L]$. One can assume $u \leftrightarrow t\; [K]$ as otherwise by (b) and (a) one of the desired conclusions $u \to w \leftarrow t \leftrightarrow v\; [K]$ is derived. Observe that $u \leftrightarrow w \leftrightarrow t \leftrightarrow v$ in K by (a). One can assume $u \to t \leftarrow v$ in K as otherwise another desired conclusion $u - t - v\; [K]$ is valid. If $t \to w$ in K then the fact that K is acyclic implies $u \to w$ in K which means $u \to w \leftarrow t \leftrightarrow v\; [K]$. If $w \to t$ then the fact $w - t - v\; [L]$ implies by (A:2) $w \leftrightarrow v\; [K]$. Moreover, one can assume that $u \not\leftrightarrow v\; [K]$ as otherwise one of the desired conclusions $u \leftrightarrow v\; [K]$ holds. This means $(u,v) \rightsquigarrow t\; [K]$ and by $(*)$ conclude that $u \perp\!\!\!\perp v \,|\, W\; [L]$ where $u, v, t \notin W \equiv pa_K(u)pa_K(v)$. Necessarily, $w \notin W$ as otherwise $u \to w \leftarrow t \leftrightarrow v$ is a path in L which is active w.r.t. W. The fact $w \notin pa_K(u)pa_K(v)$ then implies $u \to w \leftarrow v\; [K]$ which is one of the desired conclusions. $\square$

The condition (C:1) is illustrated by Figure 11. This condition makes it possible to evidence immediately that the pair of DAGs from Figure 10 does not satisfy the inclusion conditions. Indeed, one has $a \to b \leftarrow c \to d$ in L but none of the following conclusions of (C:1) is valid in K: neither $a \leftrightarrow d$ nor $a - c - d$ nor $a \to b \leftarrow d$ nor $a \to b \leftarrow c \leftrightarrow d$ in K.

Let us summarize some important necessary conditions derived in Section 5.4.

SUMMARY 1 The following graphical conditions are implied by the inclusion conditions for K in L and therefore they are necessary for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$:

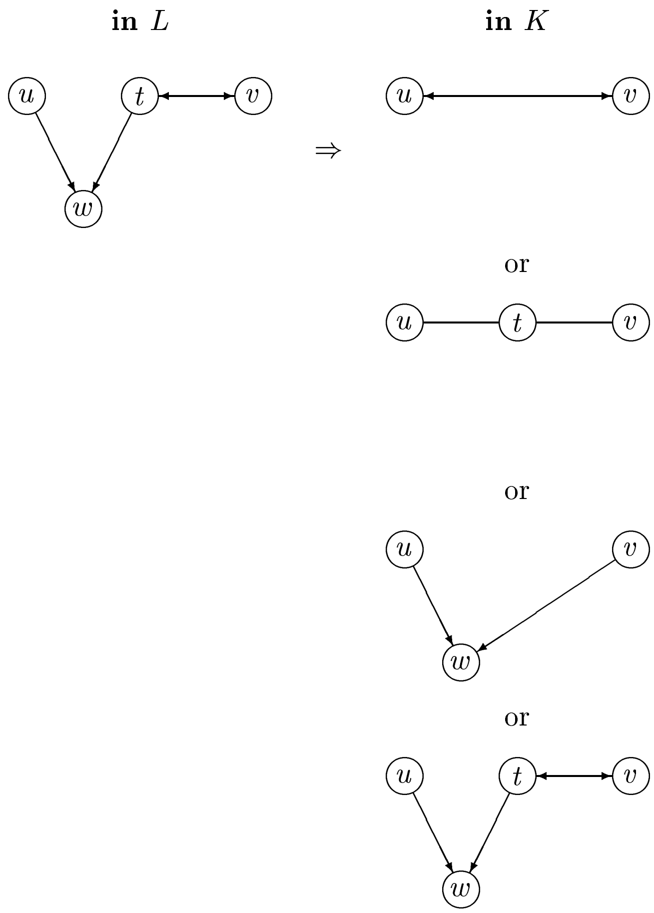

Figure 11: Illustration of the rule (C:1).

(a) $u \leftrightarrow v\ [L] \Rightarrow u \leftrightarrow v\ [K]$,

(b) $u \to w \leftarrow v\ [L] \Rightarrow u \leftrightarrow v\ [K]$ or $u \to w \leftarrow v\ [K]$,

(c) $u — w — v\ [L] \Rightarrow u \leftrightarrow w\ [K]$ or $u — w — v\ [K]$,

(d) $u \to w \leftarrow t \leftrightarrow v\ [L] \Rightarrow u \leftrightarrow v\ [K]$ or $u — t — v\ [K]$ or $u \to w \leftarrow v\ [K]$ or $u \to w \leftarrow t \leftrightarrow v\ [K]$,

(e) $u — w — t — v\ [L] \Rightarrow u \leftrightarrow v\ [K]$ or $u — w — v\ [K]$ or $u — t — v\ [K]$ or $u — w — t — v\ [K]$.

In Section 6 we will show that these conditions are also sufficient in case $|\mathcal{E}(K)| \leq |\mathcal{E}(L)| + 1$ (see Summary 2). To this end the observation that the validity of the conditions (a)-(e) is an invariant of equivalence classes of DAGs is needed.

LEMMA 5.6 Let K, L, K', L' be DAGs over N such that $\mathcal{I}(K) = \mathcal{I}(K')$ and $\mathcal{I}(L) = \mathcal{I}(L')$. Then, the validity of the conditions (a)-(e) from Summary 1 for the pair K and L is equivalent to their validity for the pair K' and L'.

Proof: This is the hint of the proof only. The basic idea is to reformulate the collection of conditions in an equivalent way with help of invariants of equivalence classes. Let us introduce for a DAG G over N and (distinct) nodes u, v, t, w the following notation.

1. $u \leftrightarrow v\ [[G]]$ means $u \leftrightarrow v\ [G]$,
2. $u \to w \leftarrow v\ [[G]]$ means $u \to w \leftarrow v\ [G]$ and $u \not\leftrightarrow v\ [G]$,
3. $u — w — v\ [[G]]$ means $u — w — v\ [G]$ and $u \not\leftrightarrow v\ [G]$,
4. $u \to w \leftarrow t \leftrightarrow v\ [[G]]$ means

   $u \to w \leftarrow t \leftrightarrow v\ [G]$, $u \not\leftrightarrow v\ [G]$, $\neg\{\, u — t — v\ [G]\,\}$ and $\neg\{\, u \to w \leftarrow v\ [G]\,\}$,
5. $u — w — t — v\ [[G]]$ means

   $u — w — t — v\ [G]$, $u \not\leftrightarrow v\ [G]$, $\neg\{\, u — w — v\ [G]\,\}$ and $\neg\{\, u — t — v\ [G]\,\}$.

Let [a] respectively [b] etc. denote the condition (a) respectively (b) etc. where the symbol $[L]$ is replaced by the symbol $[[L]]$ and the symbol $[K]$ by the symbol $[[K]]$.

The first observation is that the collection of conditions (a)-(e) is equivalent to the collection of conditions [a]-[e]. For example, (a)$\Leftrightarrow$[a] by definition, (b)$\Rightarrow$[b] follows from the conventions above and [a],[b]$\Rightarrow$(b) holds because

$$u \to w \leftarrow v\ [G] \quad \text{implies} \quad u \leftrightarrow v\ [[G]] \text{ or } u \to w \leftarrow v\ [[G]]$$

for every DAG G and nodes u, v, w. A similar principle can be used to show (c)$\Rightarrow$[c], [a],[c]$\Rightarrow$(c), (d)$\Rightarrow$[d], [a],[b],[c],[d]$\Rightarrow$(d), (e)$\Rightarrow$[e] and [a],[c],[e]$\Rightarrow$(e).

The second observation is that the conditions [a]-[e] are invariants of equivalent classes of DAGs. This is because the statements with $[[G]]$ introduced above are invariants of equivalence classes. This proposition can be derived as a consequence of Lemmas 5.2, 5.3 and 5.5. Indeed, suppose that G and G' are DAGs over N such that $\mathcal{I}(G) = \mathcal{I}(G')$. Then by Lemma 5.2 both the inclusion conditions for G in G' and the inclusion conditions for G' in G hold. Which means the conditions (a) and (b) hold both in the form with $K = G$ and $L = G'$ and in the version with $K = G'$ and $L = G$. Moreover, by Lemma 5.3 the conditions (A:2) and (A:3) hold in both versions and by Lemma 5.5 the condition (C:1) holds in both versions.

To illustrate the idea of the proof let us show that $u - w - v$ $[[G']]$ implies $u - w - v$ $[[G]]$. The assumption implies $u - w - v$ $[G']$ and by (A:2) derive that either $u \leftrightarrow v$ $[G]$ or $u - w - v$ $[G]$. In case $u \leftrightarrow v$ $[G]$ we conclude $u \leftrightarrow v$ $[G']$ by the condition (a) with $K = G'$, $L = G$. But this contradicts the assumption $u - w - v$ $[[G']]$. Thus, $u - w - v$ $[G]$ and $u \not\leftrightarrow v$ $[G]$ for the same reason. This means $u - w - v$ $[[G]]$. This procedure works also in other cases. □

5.5 Nonlocality aspect

The majority of the conditions mentioned above were local (concerning of at most 4 nodes) or at least relatively local (see the beginning of Section 5.4). In this section we show that one cannot expect full characterization of $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ in terms of graphical conditions which have the premise formulated in terms of K, the conclusion in terms of L and are relatively local.

Consider the independence model whose only non-trivial independence statement corresponds to the disjoint triplet $\langle a, b|Z\rangle$. This model is induced by a DAG K over N in which two nodes a and b are not adjacent but all the other adjacencies are present. The nodes a and b have the set Z as the set of common parents and the remaining nodes $R = N \setminus Z \cup \{a, b\}$ are children of both a and b. Figure 12 shows a DAG of this type.

Now, it is very easy to construct a DAG L such that there is only a single path from a to b in L and such that $a \top\!\!\!\top b \,|\, Z$ $[L]$. Obviously, this trail can be made as long as one likes which means that a plenty of such DAGs L exists. The significance of this example is that it shows that it is almost impossible to formulate a set of conditions that are necessary and sufficient and that consider *local* properties of the graph only. The reason is that it is impossible to check the independence statement $a \perp\!\!\!\perp b \,|\, Z$ $[L]$ by considering local properties of L: the whole trail between a and b needs to be investigated to conclude that $a \top\!\!\!\top b \,|\, Z$ $[L]$.

So, this general example shows that we have to look into a set of conditions in which at least one of the conditions has a non-local aspect. It also explains why no success has been achieved so far in formulating a set of conditions because most of them have been local conditions. This kind of property is more or less suggested by the local conditions we have to determine the equivalence of DAGs.

5.6 Necessary and sufficient condition

The last inclusion condition $(*)$ from Section 5.3 can be strengthened to get a necessary and sufficient condition.

DEFINITION 5.9 Let K and L be DAGs over a set of variables N. The following condition is called the *enforced inclusion condition*:

$$(**)\;\; a \not\leftrightarrow b\;[K] \quad \Rightarrow \quad \{a\} \perp\!\!\!\perp \{b\} \,|\, pa_K(a) pa_K(b)\;[L].$$

Natural consequence of the next lemma is that $(**)$ implies all inclusion conditions from Section 5.3 and hence the Verma's conditions from Section 5.2.

LEMMA 5.7 Let K and L be DAGs over a set of variables N. Then $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ iff the enforced inclusion condition $(**)$ holds.

Proof: If $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ then $(**)$ by Lemma 2.2. By Lemma 4.1 to show $(**) \Rightarrow \{\mathcal{I}(K) \subseteq \mathcal{I}(L)\}$ it suffices to verify $\mathcal{L}_{K,\theta} \subseteq \mathcal{I}(L)$ for an input list $\mathcal{L}_{K,\theta}$. The essential tool for proving this

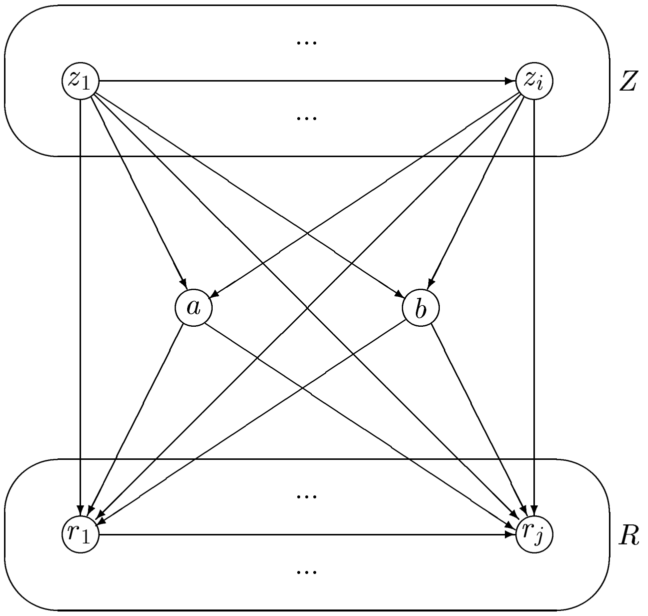

Figure 12: A counterexample to locality of conditions.

is the observation that $\mathcal{I}(L)$ is a graphoid which satisfies the composition property (see Section 4). Let us fix $u \in N$. The aim is to show that

$$\langle u, pre_\theta(u) \setminus pa_K(u) | pa_K(u) \rangle \in \mathcal{I}(L)\,. \tag{9}$$

The claim is trivial in case $pre_\theta(u) \setminus pa_K(u) = \emptyset$. Thus suppose that $V \equiv pre_\theta(u) \setminus pa_K(u) \neq \emptyset$. Let $v_1, \ldots, v_m$, $m \geq 1$ be the ordering of nodes in V according to θ. Denote $V_j \equiv \{v_1, \ldots, v_j\}$ for $j = 1, \ldots, m$ and $D \equiv pa_K(u)$. The idea is to show by induction on $j = 1, \ldots, m$ that $\langle u, V_j | D \rangle \in \mathcal{I}(L)$. This gives the desired conclusion. In case $j = 1$ the fact $u \not\leftrightarrow v_1$ $[K]$ implies by $(**)$ that $\langle u, v_1 | D \rangle = \langle u, v_1 | D \cup pa_K(v_1) \rangle \in \mathcal{I}(L)$ owing to $pa_K(v_1) \subseteq D$.

In case $j > 1$ introduce $C \equiv pa_K(v_j) \setminus D$ and observe that $C \subseteq V_{j-1}$. The fact $u \not\leftrightarrow v_j$ $[K]$ implies by $(**)$ $\langle u, v_j | CD \rangle \in \mathcal{I}(L)$. The induction hypothesis says $\langle u, V_{j-1} | D \rangle \in \mathcal{I}(L)$ which implies by decomposition $\langle u, V_{j-1} \setminus C | D \rangle \in \mathcal{I}(L)$ and $\langle u, C | D \rangle \in \mathcal{I}(L)$. The latter fact implies together with $\langle u, v_j | CD \rangle \in \mathcal{I}(L)$ by contraction $\langle u, v_j C | D \rangle \in \mathcal{I}(L)$. This together with the former fact $\langle u, V_{j-1} \setminus C | D \rangle \in \mathcal{I}(L)$ gives $\langle u, V_j | D \rangle = \langle u, v_j V_{j-1} | D \rangle \in \mathcal{I}(L)$ by composition. This concludes the proof of (9) and therefore the proof of $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. □

5.7 Meek's conjecture

Another method of characterization of inclusion is in an algorithmic fashion. This has benefit that it is closer to application in Bayesian network learning algorithms. In [8] (Chapter 4, Conjecture 22) Meek formulated a conjecture in an algorithmic form which is based on two operations on DAGs: legal arrow reversal introduced in Definition 3.1 (called covered arc-reversal by Meek) and by legal arrow adding introduced below. Note that Lemma 3.1 says that a legal arrow reversal does not affect the induced independence model.

Definition 5.10 By *legal arrow adding* we understand the change of a DAG L into a directed graph K by adding an an arrow $a \to b$ (in K) which is not an edge in L on condition that the resulting graph K is a DAG.

OBSERVATION 7 Let K, L are DAGs over N such that K is obtained from L by (legal) arrow adding. Then $\mathcal{I}(K) \subset \mathcal{I}(L)$.

Proof: The inclusion $\mathcal{D}(L) \subseteq \mathcal{D}(K)$ follows directly from the definition of d-separation: every active path in L remains an active path in K. Hence $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. The strict inclusion follows from Lemma 2.3. □

Meek's conjecture can be formulated in the following way. Given two DAGs K and L over N the conjecture says that $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ iff

(MC) there exists a sequence of DAGs $G_1, \ldots, G_n$, $n \geq 1$ such that $G_1 = L$, $G_n = K$ and the graph G_{i+1} is obtained from G_i either by legal arrow reversal or by legal arrow adding for $i = 1, \ldots, n-1$.

Of course, the condition (MC) is sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ by Lemma 3.1 and Observation 7. The difficult part is to show that (MC) is a necessary condition for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. However, the following equivalent formulation of (MC) (used in Section 1 and abstract) seems more elegant.

CONJECTURE 1 *(Meek* [8]*)*
Let K and L be DAGs over a set of variables N such that $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. Then there exists a sequence of DAGs $H_1, \ldots, H_n$ such that $K = H_1$, $L = H_n$ and the graph H_{i+1} is obtained from H_i by applying either the operation a legal arrow reversal or the operation of arrow removal for $i = 1, \ldots, n-1$.

Till these days no counterexample is known for Meek's conjecture. In fact in the following section we show that it is valid when two DAGs differ in at most one adjacency.

REMARK 5 One may think that even a simpler version of Meek's conjecture could be valid. Namely that for two DAGs K and L over N the inclusion $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ implies that there exists a sequence of DAGs $K, \ldots, K_*, \ldots, L_*, \ldots, L$ where K_* is obtained from K by a sequence of legal arrow reversals, L_* is obtained from K_* by a sequence of arrow removals and L is obtained from L_* by a sequence of legal arrow reversals. This is to warn the reader that this is not the truth. A counterexample is in Figure 13. The example shows two DAGs K and L such that there are no equivalent K_* and L_* which have the same terminal node (c is always a terminal node in K_* but not in L_*). Thus, it is not possible to get from any K_* to any L_* by arrow removals as the causal orderings of these always differs. On the other hand, $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ since L can be obtained from K by removal of $a \to c$, then legal reversal of $d \to c$ and removal of $b \to d$.

6 Partial sufficiency result

The main result of this section is that Meek's conjecture is correct in case that the graphs K and L differ in at most one adjacency. In fact, we prove even more: the graphical conditions gathered in Summary 1 (see Section 5.4.3) are necessary and sufficient both for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ and for the validity of Meek's conjecture in case that K and L differ in at most one adjacency. Recall that in case that they have the same adjacencies only three of these conditions are enough (see Lemma 5.1). The main step is to show that the conditions from Summary 1 are sufficient for validity of the condition (MC) in the considered special case.

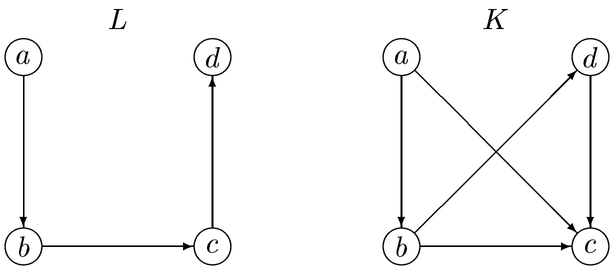

Figure 13: Meek's conjecture cannot be simplified.

LEMMA 6.1 Let K, L are DAGs over N satisfying the condition from Summary 1 and the condition

(•) $|\mathcal{E}(K)| = |\mathcal{E}(L)| + 1$.

Then there exists a sequence $G_1, \ldots, G_n$, $n \geq 2$ of DAGs over N and $1 \leq m < n$ such that

- $G_1 = L$,
- G_{i+1} is obtained from G_i by legal arrow reversal for $i = 1, \ldots, m-1$,
- $G_{m+1} \equiv K_*$ is obtained from $G_m \equiv L_*$ by legal arrow adding,
- G_{i+1} is obtained from G_i by legal arrow reversal for $i = m+1, \ldots, n-1$,
- $G_n = K$.

Proof: The proof is done by induction on the number of vertices $|N|$. If $|N| \leq 2$ then Lemma 6.1 is trivial. To verify the induction step assume that the statement of the lemma is valid for any pair of DAGs over a set of variables N' with $|N'| < |N|$.

The first step to verify its validity for N is to choose a terminal node t in K. It may happen that $t \to y$ in L for some $y \in N$. The second step is to perform a legal arrow reversals of these edges as long as this is possible. Thus, a sequence $L = G_1, \ldots, G_k$, $k \geq 1$ of DAGs over N is created where G_{i+1} is obtained from G_i by legal arrow reversal for $i = 1, \ldots, k-1$. We will use the following notation:

$$L_* = G_k, \quad P = pa_{L_*}(t), \quad C = ch_{L_*}(t), \quad X = pa_K(t) \setminus (P \cup C)\,. \tag{10}$$

The situation is illustrated in Figure 14. By Lemma 3.2 one knows that L and L_* are equivalent which means they have the same underlying graph and immoralities. Since no arrow $t \to y$ in L_* can be legally reversed at least one of the following four cases has to occur.

I. $C = \emptyset = X$,

II. $C = \emptyset$ and $X \neq \emptyset$,

III. $P \setminus pa_{L_*}(c) \neq \emptyset$ for some $c \in C$,

IV. $pa_{L_*}(c) \setminus P \cup \{t\} \neq \emptyset$ for some $c \in C$.

Indeed, the conditions in **III.** and **IV.** cover the case $pa_{L_*}(c) \neq P \cup \{t\} = pa_{L_*}(t) \cup \{t\}$ which is just the situation when legal reversal of $t \to c$ in L_* is not possible. The rest of the proof depends on the case which occurs.

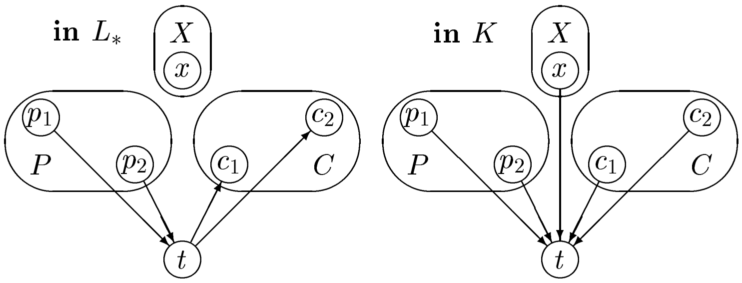

Figure 14: General starting situation.

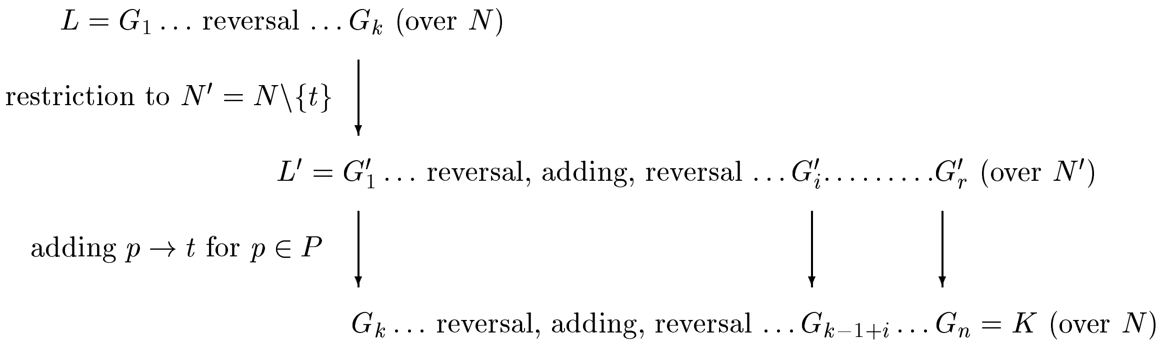

Figure 15: Schema of the proof in case **I.**

I. If $C = \emptyset = X$ then $pa_{L_*}(t) = pa_K(t) = P$. Introduce the graph L' as the induced subgraph of L_* for $N \setminus \{t\}$ and K' as the induced subgraph of K for $N \setminus \{t\}$. Observe that L' and K' are DAGs over $N' \equiv N \setminus \{t\}$. Of course, the conditions (a)-(e) for L' and K' are fulfilled as they depend on induced subgraph. Since ($\bullet$) holds for L' and K' as well by the induction hypothesis there exists a sequence $L' = G'_1, \ldots, G'_r = K'$, $r \geq 2$ of DAGs over N' satisfying required relationships (see Figure 15 for illustration). Introduce the graph G_{k-1+i} for $i = 1, \ldots, r$ as the graph over N obtained from G'_i by adding a bunch of arrows $p \to t$ for $p \in P$. It is left to the reader to verify that $G_1, \ldots, G_k, \ldots, G_{k+r-1} = G_n$ is the desired sequence of DAGs over N (the main argument is that t is a terminal node). Note that arrow adding operation can occur later in $G_k, \ldots, G_n$ which means that the 'actual' $L_* = G_m$ whose existence is claimed in Lemma 6.1 may differ from G_k.

However, in cases **II.**, **III.** and **IV.** one can put $m = k$ and define $K_* = G_{m+1}$ as the graph obtained from $L_* = G_m$ by legal adding of a certain arrow $y \to z$ in K_* which is also an edge in K. Which arrow is added depends on the considered case. However, in each of three considered cases K and K_* are shown to have the same underlying graph and immoralities in the following manner.

1. $\mathcal{E}(K) = \mathcal{E}(K_*)$
   Since L_* and L are equivalent the condition (a) gives $\mathcal{E}(L_*) = \mathcal{E}(L) \subseteq \mathcal{E}(K)$ which together with $\{y, z\} \in \mathcal{E}(K)$ and the definition of K_* implies $\mathcal{E}(K_*) \subseteq \mathcal{E}(K)$. But the assumption ($\bullet$) says $|\mathcal{E}(K)| = |\mathcal{E}(L)| + 1 = |\mathcal{E}(K_*)|$ which implies $\mathcal{E}(K_*) = \mathcal{E}(K)$.

2. $(u, v) \rightsquigarrow w\ [K_*]$ implies $(u, v) \rightsquigarrow w\ [K]$.
   First, consider the case $\neg(\{y, z\} \subseteq \{u, v, w\})$. By definition of K_* derive $(u, v) \rightsquigarrow w\ [L_*]$.

Since L and L_* are equivalent $(u,v) \rightsquigarrow w$ $[L]$. By $\mathcal{E}(K)=\mathcal{E}(K_*)$ observe $u \not\leftrightarrow v$ $[K]$. The assumption (b) in the form $(\tilde{\mathrm{b}})$ implies the desired conclusion $(u,v) \rightsquigarrow w$ $[K]$.
Second, consider the case $\{y,z\} \subseteq \{u,v,w\}$. Since $u \not\leftrightarrow v$ $[K_*]$ but $y \leftrightarrow z$ $[K]$ one has either $\{y,z\}=\{u,w\}$ or $\{y,z\}=\{v,w\}$. But u and v are interchangeable and one can assume $\{y,z\}=\{u,w\}$ without loss of generality. Since $y \to z$ in K_* it implies that $u=y$ and $w=z$. Thus to verify the condition 2. it suffices to evidence

$$(y,v) \rightsquigarrow z\ [K_*] \quad \Rightarrow \quad (y,v) \rightsquigarrow z\ [K] \quad \text{for any} \quad v \in N \setminus \{y,z\}\,. \tag{11}$$

3. $(u,v) \rightsquigarrow w$ $[K]$ implies $(u,v) \rightsquigarrow w$ $[K_*]$.
   First, consider the case $\neg(\{y,z\} \subseteq \{u,v,w\})$. By $\mathcal{E}(K)=\mathcal{E}(K_*)$, the definition of K_* and $\mathcal{E}(L)=\mathcal{E}(L_*)$ deduce $u \leftrightarrow w$ $[L]$ and $v \leftrightarrow w$ $[L]$. The assumption (c) in the form $(\tilde{\mathrm{c}})$ therefore implies $(u,v) \rightsquigarrow w$ $[L]$. By equivalence of L and L_* derive $(u,v) \rightsquigarrow w$ $[L_*]$ and by the definition of K_* the desired conclusion $(u,v) \rightsquigarrow w$ $[K_*]$.
   Second, consider the case $\{y,z\} \subseteq \{u,v,w\}$. Assumptions $u \not\leftrightarrow v$ $[K]$ and $y \leftrightarrow z$ $[K]$ necessitate $\{y,z\} \neq \{u,v\}$ and one can assume without loss of generality that $\{y,z\} = \{u,w\}$. This essentially includes two subcases: $y=u$, $z=w$ (if $y \to z$ in K) and $y=w$, $z=u$ (if $z \to y$ in K). Therefore, to verify 3. it suffices to evidence the condition

$$(y,v) \rightsquigarrow z\ [K] \quad \Rightarrow \quad (y,v) \rightsquigarrow z\ [K_*] \quad \text{for any} \quad v \in N \setminus \{y,z\}\,. \tag{12}$$

   and the condition

$$(z,v) \rightsquigarrow y\ [K] \quad \Rightarrow \quad (z,v) \rightsquigarrow y\ [K_*] \quad \text{for any} \quad v \in N \setminus \{y,z\}\,. \tag{13}$$

   Note that the latter condition (13) can be omitted if one is sure that $y \to z$ in K.

As soon as equivalence of K_* and K is verified one can use Lemma 3.2 to prove the existence of a sequence $K_* = G_{m+1}, \ldots, G_n = K$ of DAGs satisfying the required conditions. This will conclude the proof.

II. If $C=\emptyset \neq X$ then choose $x \in X$ and define K_* as the graph obtained from L_* by adding the arrow $x \to t$. Note for explanation that $(\bullet)$ implies that $X=\{x\}$ so that no actual choice of x is made. Since t is a terminal node in L_* in this case (see Figure 16 for illustration) the resulting graph is acyclic. That means, K_* is obtained from L_* by legal arrow adding. One also knows that $x \to t$ in K which means that to show that K_* and K are equivalent it suffices to verify (11) and (12) for $y=x$ and $z=t$. To verify (11) assume $(x,v) \rightsquigarrow t$ $[K_*]$ for $v \in N \setminus \{x,t\}$. The facts that $\mathcal{E}(K)=\mathcal{E}(K_*)$ and t is a terminal node in K then imply $(x,v) \rightsquigarrow t$ $[K]$. For verification of (12) use the same argument with K_* replaced by K.

III. If $P \backslash pa_{L_*}(c) \neq \emptyset$ for some $c \in C$ then fix one of these $c \in ch_{L_*}(t)$ and choose $p \in P \backslash pa_{L_*}(c)$. Necessarily $p \not\leftrightarrow c$ $[L_*]$ as otherwise $p \to t \to c \to p$ forms a directed cycle in L_*. Let us summarize the assumed situation as follows:

$$\exists\, p,c \in N \quad p \to t \text{ in } L_*\,, \quad t \to c \text{ in } L_* \quad \text{and} \quad p \not\leftrightarrow c \text{ in } L_*\,. \tag{14}$$

The first step is an observation that $p \leftrightarrow c$ $[K]$. Indeed, (14) and the fact that L and L_* are equivalent implies $p \leftrightarrow t \leftrightarrow c$ in L. Hence by (a) and the fact that t is a terminal node in K derive $p \to t \leftarrow c$ in K. Suppose $p \not\leftrightarrow c$ $[K]$ which implies the existence of an immorality $(p,c) \rightsquigarrow t$ $[K]$. Then $(p,c) \rightsquigarrow t$ $[L]$ by (c) in the form $(\tilde{\mathrm{c}})$ and $(p,c) \rightsquigarrow t$ $[L_*]$ by the fact that L and L_* are equivalent. The fact $c \to t$ in L_* then contradicts (14).

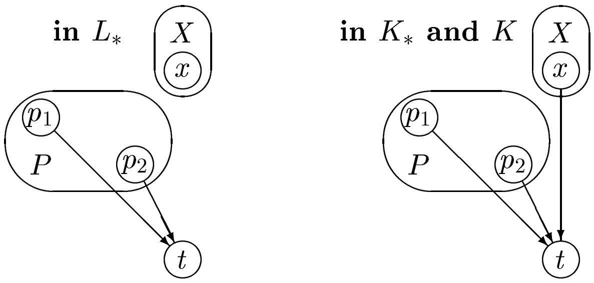

Figure 16: Situation in case **II.**

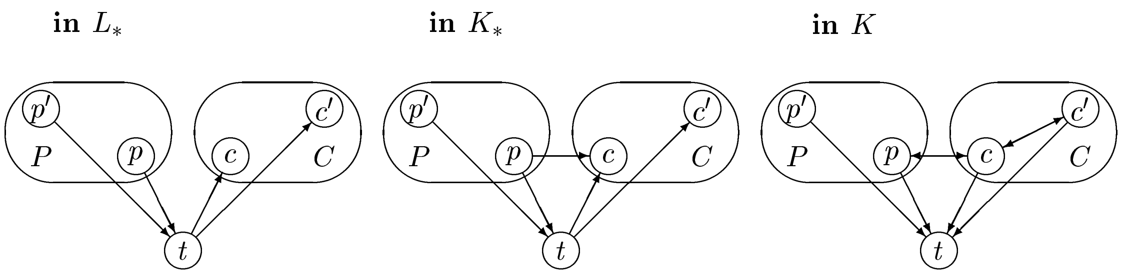

Figure 17: The situation in case **III.**

The second step is to define K_* as the graph obtained from L_* by adding the arrow $p \to c$. The situation is illustrated in Figure 17. Note for explanation that the assumption ($\bullet$) and the above mentioned consideration also imply that $p' \leftrightarrow c'$ $[L]$ for every pair of distinct nodes p', c' where $p' \in P \cup C$, $c' \in C$ and $(p', c') \neq (p, c)$. However, this is not depicted in Figure 17 for sake of lucidity. The reason is that we give general instructions how to handle the case **III.** even without assuming ($\bullet$) in this proof.

K_* is acyclic since L_* is acyclic and the arrow $p \to c$ in K_* can be (in a hypothetical directed cycle) replaced by directed path $p \to t \to c$ in L_*. Thus, K_* is obtained from L_* by legal arrow adding. As explained before **II.** to show that K_* and K are equivalent one has to verify (11), (12) and (13) for $y = p$ and $z = c$.

To verify (11) assume $(p, v) \rightsquigarrow c$ $[K_*]$ for some $v \in N \setminus \{p, c\}$. Necessarily $v \neq t$ since $p \leftrightarrow t$ $[K_*]$ but $p \not\leftrightarrow v$ $[K_*]$. First observation is $v \to c$ in L_* which follows from the definition of K_* and the fact $p \neq v$. Thus, $v \to c \leftarrow t \leftarrow p$ in L_* by (14). Since L and L_* are equivalent by Lemma 5.6 conclude that the condition (d) from Summary 1 with L_* in place of L is valid. Apply this condition to derive that one of these four conditions holds:

$$v \leftrightarrow p\ [K] \ \text{ or } \ v - t - p\ [K] \ \text{ or } \ v \to c \leftarrow t \leftrightarrow p\ [K] \ \text{ or } \ v \to c \leftarrow p\ [K]\,.$$

But $\mathcal{E}(K) = \mathcal{E}(K_*)$ implies $v \not\leftrightarrow p$ $[K]$ and the other two options are excluded as t is a terminal node in K. Thus necessarily $(p, v) \rightsquigarrow c$ $[K]$.

To verify (12) assume $(p, v) \rightsquigarrow c$ $[K]$ for $v \in N \setminus \{p, c\}$. Suppose (by contradiction) that $(p, v) \rightsquigarrow c$ is not an immorality in K_*. Since $\mathcal{E}(K) = \mathcal{E}(K_*)$ and $p \to c$ in K_* it is equivalent to the requirement $c \to v$ in K_* (see Figure 18 for illustration). It implies $c \to v$ in L_* by the definition of K_* and the fact $p \neq v$. It follows from (14) that $p \to t \to c \to v$ is a path in L_* which means $p - t - c - v$ $[L_*]$. As L and L_* are equivalent by Lemma 5.6 conclude

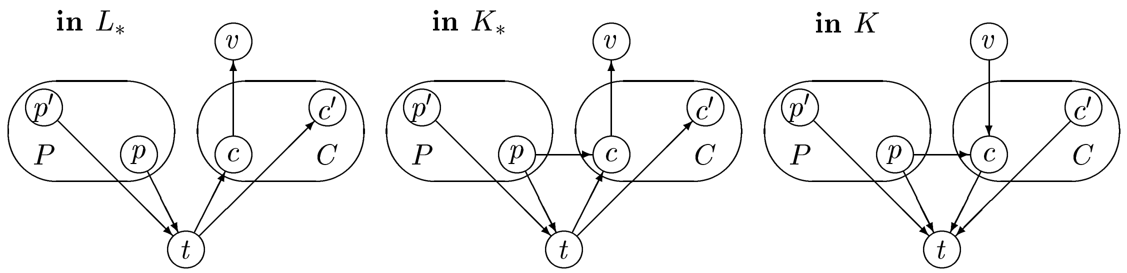

Figure 18: Proof $(p,v) \rightsquigarrow c\ [K] \;\Rightarrow\; (p,v) \rightsquigarrow c\ [K_*]$ in case **III.**

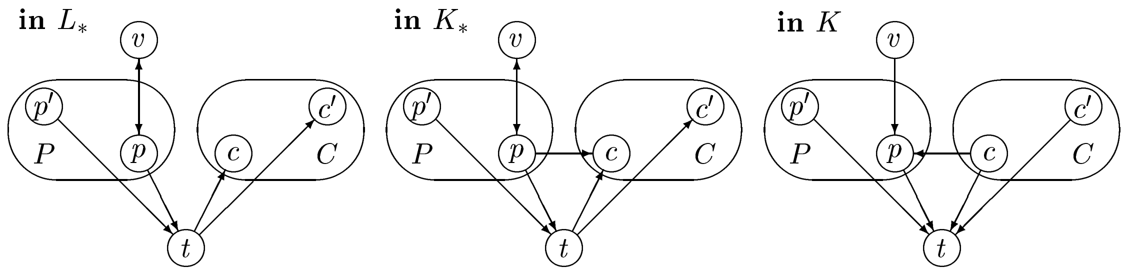

Figure 19: Immorality $(c,v) \rightsquigarrow p$ is not in K in case **III.**

that the condition (e) from Summary 1 with L_* replaced by L holds. Derive that one of these 4 cases occurs:

$$p \leftrightarrow v\ [K] \ \text{ or } \ p \,—\, c \,—\, v\ [K] \ \text{ or } \ p \,—\, t \,—\, v\ [K] \ \text{ or } \ p \,—\, t \,—\, c \,—\, v\ [K]\,.$$

The first two conditions contradicts the assumption $(p,v) \rightsquigarrow c\ [K]$ and the last two conditions the fact that t is a terminal node in K. Thus necessarily $(p,v) \rightsquigarrow c\ [K_*]$.

To show (13) it suffices to verify that $(c,v) \rightsquigarrow p$ is never immorality in K for any $v \in N \setminus \{p,c\}$. Indeed, suppose $(c,v) \rightsquigarrow p\ [K]$ for contradiction (see Figure 19 for illustration). The fact $\mathcal{E}(K_*) = \mathcal{E}(K)$ implies $p \leftrightarrow v\ [K_*]$. The definition of K_* and $v \neq c$ then gives $p \leftrightarrow v\ [L_*]$. By (14) $c \leftarrow t \leftarrow p$ in L_* which means that $c \,—\, t \,—\, p \,—\, v\ [L_*]$. Again by the condition (e) where L_* plays the role of L derive that one of the following fours conditions holds:

$$c \,—\, t \,—\, p \,—\, v\ [K] \ \text{ or } \ c \,—\, t \,—\, v\ [K] \ \text{ or } \ c \leftrightarrow v\ [K] \ \text{ or } \ c \,—\, p \,—\, v\ [K]\,.$$

The first two conditions cannot occur since t is a terminal node in K and both remaining conditions exclude $(c,v) \rightsquigarrow p\ [K]$.

IV. If $pa_{L_*}(c') \setminus P \cup \{t\} \neq \emptyset$ for all $c' \in C$ and the case **III.** does not occur then choose such $c \in C$ which has no other ancestor in C, that is $an_{L_*}(c) \cap C = \{c\}$. Since **III.** is excluded necessarily $pa_{L_*}(c) \setminus P \cup \{t\} \neq \emptyset$ and one can choose a node x from this set. Observe that $\neg(t \to x \text{ in } L_*)$ since otherwise $x \in C$ and $x \to c$ in L_* which contradicts the choice of c. Let us summarize the assumed situation as follows:

$$\exists\, x, c \in N \quad x \neq t \quad x \to c \text{ in } L_*\,, \quad t \to c \text{ in } L_* \ \text{ and } \ x \not\leftrightarrow t \text{ in } L_*\,. \tag{15}$$

The first step is to observe that $x \to t$ in K. Indeed, (15) says $(x,t) \rightsquigarrow c\ [L_*]$ and hence $(x,t) \rightsquigarrow c\ [L]$. Because $c \to t$ in K the assumption (b) in the form ($\tilde{\text{b}}$) implies $x \leftrightarrow t\ [K]$. Since t is a terminal node in K one has $x \to t$ in K.

The second step is to define K_* as the graph obtained from L_* by adding the arrow $x \to t$. The situation is shown in Figure 20. One can show that K_* is acyclic by contradiction. Indeed, in case there exists a directed cycle $x \to t \to d \to \ldots \to x$ in K_* one can distinguish two subcases. If $d = c$ then a directed cycle $x \to c = d \to \ldots \to x$ is in L_* which contradicts acyclicity of L_*. If $d \neq c$ then $t \to d$ in L_* says $d \in C$ and $d \to \ldots \to x$ in L_* implies with (15) that $d \in an_{L_*}(c)$ which contradicts the choice of c. Thus, K_* is acyclic which means it was obtained from L_* by legal arrow adding. As explained before **II.** to show that K_* and K are equivalent one has to verify (11) and (12) for $y = x$ and $z = t$.

To verify (11) assume $(x,v) \rightsquigarrow t\ [K_*]$ for some $v \in N \setminus \{x,t\}$. Since $\mathcal{E}(K) = \mathcal{E}(K_*)$ the fact that t is a terminal node in K implies the desired conclusion $(x,v) \rightsquigarrow t\ [K]$.

To verify (12) assume $(x,v) \rightsquigarrow t\ [K]$ for some $v \in N \setminus \{x,t\}$. The essential fact is that $v \in P = pa_{L_*}(t)$. To evidence it observe $v \leftrightarrow t\ [L_*]$ by $\mathcal{E}(K_*) = \mathcal{E}(K)$ and the definition of K_* ($v \neq x$). Suppose for contradiction that $v \in C = ch_{L_*}(t)$. The case $v = c$ is excluded since $x \leftrightarrow c\ [L_*]$ but $x \not\leftrightarrow v\ [L_*]$ by the condition (a). The first observation is that $c \leftrightarrow v\ [K]$ since otherwise $(c,v) \rightsquigarrow t\ [K]$ which implies $(c,v) \rightsquigarrow t\ [L]$ by the assumption (c) in the form ($\tilde{\text{c}}$) and $(c,v) \rightsquigarrow t\ [L_*]$ by equivalence of L and L_* which contradicts (15). Thus $c \leftrightarrow v\ [K]$ and therefore $c \leftrightarrow v\ [L_*]$ by $\mathcal{E}(K) = \mathcal{E}(K_*)$ and the definition of K_*. The obtained situation is shown in Figure 21. The second observation is that $(x,v) \rightsquigarrow c$ is not an immorality in K as otherwise by the assumption (c) in the form ($\tilde{\text{c}}$) and (15) it is an immorality in L and therefore in L_*. This means $v \to c$ in L_* which together with the fact $v \in C$ contradicts the choice of c. Thus $(x,v) \rightsquigarrow c$ is not an immorality in K, i.e. one has $x - c - v\ [K]$ - see Figure 21. Observe that $x \to c \leftarrow t \to v$ in L_* by (15). As L and L_* are equivalent the condition (d) with L replaced by L_* (see Lemma 5.6) implies that one of these four conditions holds:

$$x \leftrightarrow v\ [K] \quad \text{or} \quad x \to c \leftarrow v\ [K] \quad \text{or} \quad x - t - v\ [K] \quad \text{or} \quad x \to c \leftarrow t \leftrightarrow v\ [K]\,.$$

The first option is excluded by $(x,v) \rightsquigarrow t\ [K]$, the second option was excluded above and the last two options contradict the fact that t is a terminal node in K. Thus the assumption $v \in C$ leads to contradiction which means $v \in P$. If $v \in P$ one has $v \to t$ in L_* and therefore by definition of K_* also $v \to t$ in K_* and $x \to t$ in K_*. Since $\mathcal{E}(K_*) = \mathcal{E}(K)$ it implies the desired fact $(x,v) \rightsquigarrow t\ [K_*]$. This concludes the proof of (12).

Thus the case **IV.** is shown, which concludes the proof. □

REMARK 6 It seems strange that the condition (B:2) which was shown in Section 5.4.2 to be a necessary condition for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ (even in case of validity of ($\bullet$) - see Figure 7) was not involved in Summary 1 and was not used in the proof of Lemma 6.1. The reason is that under assumption ($\bullet$) it follows from the conditions (a)-(e). However, this is not the case in general.

The consequence of Lemma 6.1 is that the inclusion conditions are sufficient for validity of Meek's conjecture in the considered special case.

CONSEQUENCE 6.1 Let K, L be DAGs over N such that the inclusion conditions (a), (b), ($*$) for K in L and the condition ($\bullet$) hold. Then the conclusion of Lemma 6.1 hold.

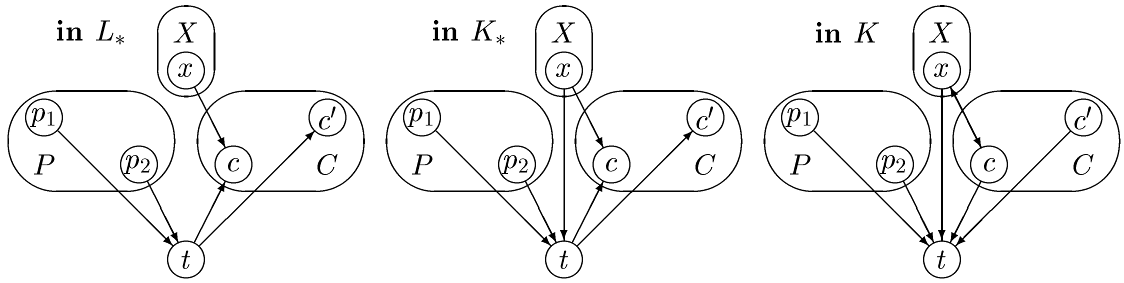

Figure 20: Situation in case **IV.**

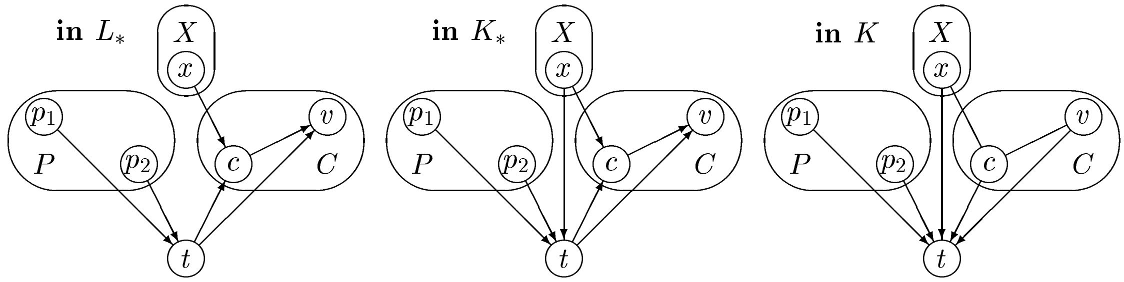

Figure 21: The case $(x, v) \rightsquigarrow t\ [K]$; the proof of the essential fact $v \in P$ by contradiction.

Proof: Combine Summary 1 and Lemma 6.1. □

Let us summarize the results.

SUMMARY 2 Let K and L are DAGs over N such that $|\mathcal{E}(K)| \leq |\mathcal{E}(L)| + 1$. Then the following four conditions are equivalent:

(i) $\mathcal{I}(K) \subseteq \mathcal{I}(L)$,

(ii) the inclusion conditions for K in L from Definition 5.1 hold,

(iii) five graphical conditions gathered in Summary 1 hold,

(iv) the condition (MC) from Section 5.7 holds.

Proof: Lemma 5.2 gives **(i)** $\Rightarrow$ **(ii)**; the implication **(ii)** $\Rightarrow$ **(iii)** follows from Summary 1 (which follow from Lemmas 5.3 and 5.5). Lemma 6.1 gives **(iii)** $\Rightarrow$ **(iv)**. The implication **(iv)** $\Rightarrow$ **(i)** can be verified by repetitive application of Lemma 3.1 and Observation 7. □

7 Conjectures

In this section we gather various conjectures.

7.1 Inclusion conditions

The following conjecture was already mentioned in Section 5.3.

Conjecture 2 *(Bouckaert)*
The conditions (a),(b) and ($*$) from Section 5.3 are necessary and sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$.

So, we conjecture that the inclusion conditions are sufficient as well as necessary for inclusion.

7.2 Graphical necessary conditions

This conjecture follows the ideas which are behind the results from Section 5.4. The basic idea is that every dependence complex in L has a subcomplex in K. But one has carefully determine which dependence complexes in K are supposed to be subcomplexes of a given dependence complex in L.

The set of nodes of a complex κ in a DAG L over N between nodes a and b for a set $C \subseteq N \setminus \{a,b\}$ can be partitioned into several pairwise disjoint subsets. The nodes belonging to respective active path π are divided by its collider nodes into *open areas*. Every collider node d of π has its *collider area* which is only the node d in case $d \in C$ or the set of nodes of respective rope $\rho(d)$ in κ in case $d \notin C$. By the *root* of a collider area is understood the only node in C within this area. In case π has no collider node the nodes of κ form one open area only. The areas are 'ordered': after the open area which contains a a collider area follows, then another open area etc., the last area is the open area containing b.

Subcomplexes of κ should correspond to certain subsets A of the set of nodes of κ, namely sets A such that

- $\{a,b\} \subseteq A$,
- when A intersects a collider area then it contains the root of the area,
- when A intersects two different collider areas then it intersects at least one open area between them.

By a subcomplex of κ in a DAG K over N determined by A can be understood a dependence complex λ in K which has A as the set of nodes and whose active path and ropes are determined by the requirement that every collider area of λ is a subset of a collider area of κ and the order of nodes in λ 'follows' the order in π and the ropes of κ. An example is the concept of shortening (of an open path) from Section 5.4.1 or the concept of subconfiguration from Section 5.4.2.

Conjecture 3 *(Studený)*
The following condition

(D) Every dependence complex in L has a subcomplex in K,

is necessary and sufficient for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$.

The problem is to verify necessity of this general condition, namely that it is implied by the inclusion conditions (a), (b) and ($*$) - which is also a conjecture. The sufficiency for $\mathcal{I}(K) \subseteq \mathcal{I}(L)$ is quite clear because of Lemma 2.1.

Remark 7 Intuitive meaning of conditions (A), (B) and (D) is as follows. Whenever $u = w_1 — \ldots — w_n = v$, $n \geq 2$ is an open path in a DAG G then for $S \supseteq \{w_i;\, 1 < i < n\}$ one has $u \not\perp\!\!\!\perp v \mid -S\ [G]$ (see Observation 3). Note that converse implication need not hold for every minimal $S \subseteq N \setminus \{u,v\}$ with $u \not\perp\!\!\!\perp v \mid -S\ [G]$ as an example in Figure 22 shows. However, the condition (A) more or less corresponds to the implication $u \not\perp\!\!\!\perp v \mid -S\ [L] \Rightarrow u \not\perp\!\!\!\perp v \mid -S\ [K]$.

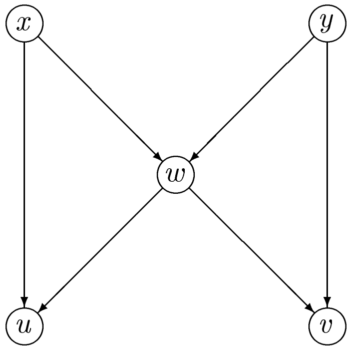

Figure 22: No open path between u and v through $S = \{x, y\}$ exists in G but $u \perp\!\!\!\perp v \,|\, -S \; [G]$.

Analogously, the requirement $(u,v) \mapsto w_1, \ldots, w_n \; [G]$ implies by Observation 6 the fact $u \perp\!\!\!\perp v \,|\, +w_n \; [G]$. Thus, the condition (B) has meaning of implication $u \perp\!\!\!\perp v \,|\, +w_n \; [L] \Rightarrow u \perp\!\!\!\perp v \,|\, +w_n \; [K]$.

Finally, let κ be a complex between u and v for C in L, T is the set of its nodes in C and S is the set if its nodes outside $C \cup \{u, v\}$. Then one has $u \perp\!\!\!\perp v \,|\, +T-S \; [L]$. The conjectured condition (D) then corresponds more or less to the implication $u \perp\!\!\!\perp v \,|\, +T-S \; [L] \Rightarrow u \perp\!\!\!\perp v \,|\, +T-S \; [K]$.

7.3 Extension of Meek's conjecture

The way in which we proved the validity of Meek's conjecture in the considered special case indicates possible method of its verification in the general case. The conjecture says something about the existence of a sequence of DAGs only; we specify the conjecture by saying how to generate this sequence. We believe that the procedure described below could lead to the general proof of Meek's conjecture.

CONJECTURE 4 *(Kočka)*
Let K and L arer DAGs over N such that $\mathcal{I}(K) \subseteq \mathcal{I}(L)$. Then following algorithm will convert L into K (it modifies L till $L = K$):

1. Let Y denote the set of nodes which were already processed and put $Y = \emptyset$.

2. Choose a terminal node in the induced subgraph $K_{N \setminus Y}$ and denote it by t. Continue by Step 3.

3. Legally reverse (in L) all possible arrows having t as a tail node in $L_{N \setminus Y}$. Denote the set of all children of t in $L_{N \setminus Y}$ by C and the set of parents of t in $L_{N \setminus Y}$ by P. Continue by Step 4.

4. For every $c \in C$ do this: if $P \setminus pa_L(c) \neq \emptyset$ then for every $p \in P \setminus pa_L(c)$ add (legally) the arrow $p \to c$ in L. If some arrows were added in this step then go to Step 3 otherwise continue by Step 5.

5. Put $C_* = \{c \in C\,;\, an_L(c) \cap C = \{c\}\}$. If $C_* \neq \emptyset$ then choose $c' \in C_*$, choose $x \in pa_L(c') \setminus P \cup \{t\}$, add the arrow $x \to t$ in L and go to the Step 3. If $C_* = \emptyset = C$ then continue by Step 6.

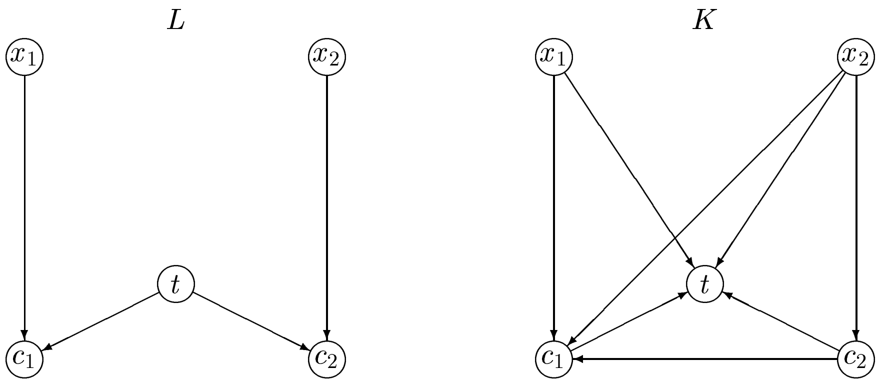

Figure 23: The simplest example of a weak point in Kočka's conjecture.

6. For every $x \in pa_K(t) \setminus P$ add the arrow $x \to t$ in L. Put $Y = Y \cup \{t\}$. If $Y \subset V$ then continue by Step 2, otherwise halt. Now one has $L = K$.

REMARK 8 All parts of this conjecture can be proved in a similar way as in the proof of Lemma 6.1 except for Step 5. The weak point of this conjecture is at this step: we believe that it is always possible to choose some c' but we don't know a method how to choose it in general. That means we do not know it in case the set C_* has more than one node. The simplest example of this situation is in Figure 23. In this case one has to choose $c' = c_2$ since $x_1 \perp\!\!\!\perp c_2 \,|\, x_2 \; [L]$ is dictated by enforced inclusion condition $(**)$ and the algorithm described in Conjecture 4 leads to the end.

7.4 Sandwich lemma conjecture

The proof of Meek's conjecture in the considered special case reduces the rest of the proof of general conjecture. We call the remaining part of the conjecture sandwich lemma conjecture, because it conjectures the existence of a DAG in between all two DAGs which differ in more then one edge. However owing to our result it is equivalent to Conjecture 1.

CONJECTURE 5 *(sandwich lemma conjecture)*
Let K and L are DAGs over N such that $\mathcal{I}(K) \subset \mathcal{I}(L)$ and $|\mathcal{E}(L)| + 1 < |\mathcal{E}(K)|$. Then there exists a DAG G over N such that $\mathcal{I}(K) \subset \mathcal{I}(G) \subset \mathcal{I}(L)$ and $|\mathcal{E}(L)| < |\mathcal{E}(G)| < |\mathcal{E}(K)|$.

8 Conclusion

The main achievements of this report are various characterizations of inclusion of DAG models and the proof of Meek's conjecture for DAGs that differ in at most one adjacency. As a warming up a rigorous proof of equivalence of DAG models (which can be considered as inclusion where the number of edges does not differ) is given.

Furthermore, we established some intuition in the characterization of inclusion and showed that all characterizations must have a non-local component. We suggested various strategies on how to attack Meek's conjecture for the general case when DAGs differ in more than one adjacency.

In the future, we would like to prove sufficiency of the characterization of inclusion using the inclusion conditions. Furthermore, the Meek's conjecture still stands to be proved for the general case.

References

[1] M. D. Chickering: A transformational characterization of equivalent Bayesian networks, in Uncertainty in Artificial Intelligence 11 (P. Besnard, S. Hanks eds.), Morgan Kaufmann 1995, pp. 87-98.

[2] M. D. Chickering: Learning equivalence classes of Bayesian network structures, in Uncertainty in Artificial Intelligence 12 (E. Horvitz, F. Jensen eds.), Morgan Kaufmann 1996, pp. 150-157.

[3] P. Giudici, R. Castelo: Markov Chain Monte Carlo model selection for DAG models, technical report 117, University of Pavia, Department of Political Economy and Quantitative Methods, September 2000.

[4] M. Frydenberg: The chain graph Markov property, Scandinavian Journal of Statistics 17 (1990), pp. 333-353.

[5] D. Heckerman, D. Geiger, M. D. Chickering: Learning Bayesian networks, the combination of knowledge and statistial data, technical report MSR-TR-94-09, Microsoft, March 1994, revised February 1995.

[6] S. L. Lauritzen, A. P. Dawid, B. N. Larsen, H.-G. Leimer: Independence properties of directed Markov fields, Networks 20 (1990), pp. 491-505.

[7] F. Matúš: Conditional independence structures examined via minors, Annals of Mathematics and Artificial Intelligence 21 (1997), pp. 99-128.

[8] C. Meek: Graphical models, selecting causal and statistical models, PhD thesis, Carnegie Mellon University, 1997.

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

[10] M. Studený: Description of structures of conditional independence by means of faces and imsets, 3rd part: examples of use and appendices, International Journal of General Systems 23 (1995), pp. 323-341.

[11] M. Studený: On recovery algorithm for chain graphs, International Journal of Approximate Reasoning 17 (1997), pp. 265-293.

[12] M. Studený: On mathematical description of probabilistic conditional independence structures, a survey paper in preparation.

[13] T. Verma, J. Pearl: Influence diagrams and d-separation, technical report 880052 (R-101), UCLA, Cognitive Systems Laboratory, March 1988.

[14] T. Verma, J. Pearl: Causal networks, semantics and expressiveness, in Uncertainty in Artificial Intelligence 4 (R. D. Schachter, T. S. Lewitt, L. N. Kanal, J. F. Lemmer eds.), North-Holland 1990, pp. 69-76.

[15] T. Verma, J. Pearl: Equivalence and synthesis of causal models, in Uncertainty in Artificial Intelligence 6 (P. P. Bonissone, M. Henrion, L. N. Kanal, J. F. Lemmer eds.), Elsevier 1991, pp. 220-227.