A graphical characterization of the largest chain graphs*

Martin Volf[†1] Milan Studený[‡2]

[1]Czech Technical University Prague, Faculty of Nuclear Sciences and Physical Engineering, Department of Mathematics, Trojanova 13, Prague 12000, Czech Republic.

[2]Institute of Information Theory and Automation, Academy of Sciences of Czech Republic, Pod vodárenskou věží 4, Prague 18208, Czech Republic, and Laboratory of Intelligent Systems, University of Economics Prague, Ekonomická 957, Prague 14800, Czech Republic.

Abstract

The paper presents a graphical characterization of the largest chain graphs which serve as unique representatives of classes of Markov equivalent chain graphs. The characterization is a basis for an algorithm constructing, for a given chain graph, the largest chain graph equivalent to it. The algorithm was used to generate a catalog of the largest chain graphs with at most five vertices. Every item of the catalog contains the largest chain graph of a class of Markov equivalent chain graphs and an economical record of the induced independency model.

Keywords: graphical models, conditional independence, independency models, chain graphs, Markov equivalence, the largest chain graph, complex arrows, protected arrows.

1 Introduction

The topic of this paper are chain graph models of conditional independence structures. The class of chain graphs was introduced by Lauritzen and Wermuth [10] as a graphical tool which allows one to represent both symmetric associations and directional influences among variables. The symmetric associations correspond to lines (= undirected edges) and the directional influences correspond to arrows (= directed edges). Note that the original research report [10] was later modified and became a basis of the paper [11]. Mathematical theory of chain graphs was developed mainly by Frydenberg [8]. The class of Markovian distributions with respect to a chain graph was introduced by means of a *moralization criterion* there, see also [12]. Moreover, Frydenberg [8] characterized Markov equivalent chain graphs (i.e., graphs inducing the same class of Markovian distributions) in graphical terms and showed that every equivalence class contains a distinguished representative which is called the *largest chain graph*.

Several later works dealt with chain graphs, for example [20], [5], [4], [16], [19], [1]. An equivalent *separation criterion* for chain graphs was introduced in [3]. It made it possible to confirm the conjecture from [8] that the global Markov condition is the strongest possible one - see [18]. Chain graphs became a topic of books as well - see [21], [13]. Cox and Wermuth [6] introduced a wider class of *joint-response chain graphs* in which two additional types of relationships among variables are considered (they are represented by dashed lines and arrows). An alternative Markov property for joint-response chain graphs with dashed arrows and solid lines was developed by Andersson, Madigan and Perlman [2], for comparison see [15].

Nevertheless, this paper is concerned with the original chain graphs (with solid lines and arrows) treated by Frydenberg [8]. One of Frydenberg's open questions was to find a procedure that for a given chain graph constructs the largest chain graph with the same Markov properties. The pool-component procedure from [17] is an example of such a procedure. In this paper we present even a more elegant solution of the problem. We give a simple direct graphical characterization of

*This work was supported by the grants GAČR n. 201/98/0478, GAAVČR n. A1075801, GAAVČR n. K1075601 and MŠMT n. VS96008.

†Corresponding author, Jiráskova 372, Dobrovice 29441, Czech Republic, e-mail: martin.volf@datatrans.cz

‡E-mail: studeny@utia.cas.cz

those chain graphs which are the largest chain graphs of (some) classes of Markov equivalent chain graphs. The characterization leads immediately to another algorithm for finding the largest chain graph which is Markov equivalent to a given chain graph.

The next section deals with basic concepts and their relevant properties. In the third section we introduce the concept of *protected arrow*. The main result of the paper says that a chain graph is the largest chain graph (of a class of Markov equivalent chain graphs) iff every its arrow is protected. The third section also contains the description of the above mentioned algorithm. In the fourth section, we used the algorithm to generate a catalog of the largest chain graphs over at most five vertices by a computer. The results of the paper and further prospects are discussed in Conclusions (the last section).

2 Basic concepts

2.1 Graphs and routes

A *hybrid graph* over V is an ordered pair $G = (V, E)$, where V is a finite non-empty set, elements of which are called *vertices* of G, and E is a set of ordered pairs of distinct vertices of G. An ordered pair (u, v) of vertices of G is called an *edge* in G, iff $(u, v) \in E$ or $(v, u) \in E$. An edge (u, v) in G is called an *undirected edge* if $(u, v) \in E$ and $(v, u) \in E$, a *directed edge* if $(u, v) \in E$ and $(v, u) \notin E$, and a *reverse directed edge* if $(u, v) \notin E$ and $(v, u) \in E$. We also use the phrases *line*, *arrow*, *reverse arrow* in G and the notation $u — v$, $u \to v$, $u \leftarrow v$ respectively. Note that our definition implies that at most one edge occurs for every ordered pair of distinct vertices. Let us give an example of a hybrid graph. Put $V = \{a, b, c\}$, $E = \{(a, b), (b, a), (a, c)\}$ and $G = (V, E)$. Then (a, b) and (b, a) are lines in G, (a, c) is an arrow in G and (c, a) is a reverse arrow in G. The pairs (b, c) and (c, b) are not edges in G. The graph G is shown in the left picture of Figure 1.

Figure 1: Examples of hybrid graphs.

Let $G = (V, E)$ be a graph over V and $U \subseteq V$ is non-empty. The graph $(U, E \cap (U \times U))$ is called the *subgraph* of G *induced* by U and denoted by G_U. A graph which contains no arrow is called *undirected*, a graph which contains no line is called *directed*. In particular, the graphs without edges are both directed and undirected graphs. The *underlying graph* of a graph G is an undirected graph obtained from G by replacing all edges in G by lines.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two hybrid graphs. We say that they are *isomorphic* if there exists a one-to-one mapping ι from V_1 to V_2 such that, for every ordered pair (u, v) of distinct vertices of G_1, $(u, v) \in E_1$ iff $(\iota(u), \iota(v)) \in E_2$. For example, the graph in the right picture of Figure 1 is isomorphic to the graph in the left picture of Figure 1. Here $\iota(a) = c$, $\iota(b) = d$, $\iota(c) = e$.

A *route* from a vertex u_1 to a vertex u_n ($n \geq 1$) in a hybrid graph G is a finite sequence $(u_1, \ldots, u_n)$ of its vertices such that (u_i, u_{i+1}) is an edge in G for all $i = 1, \ldots, n-1$. A *path* is a route which consists of distinct vertices. A *pseudo-cycle* is a route $(u_1, \ldots, u_n)$ such that $n \geq 4$ and $u_1 = u_n$. A *cycle* is a pseudo-cycle $(u_1, \ldots, u_n)$ such that $(u_1, \ldots, u_{n-1})$ is a path. A route $(u_1, \ldots, u_n)$ is called *undirected*, if $u_i — u_{i+1}$ for all $i = 1, \ldots, n-1$. It is called *descending* if either $u_i \to u_{i+1}$ or $u_i — u_{i+1}$ for all $i = 1, \ldots, n-1$. A descending route $(u_1, \ldots, u_n)$ is called *directed* if $u_j \to u_{j+1}$ for at least one $j \in \{1, \ldots, n-1\}$.

Example. Let us give a few examples of different types of routes in the graph from Figure 2:

- (a, b, c, f, g, b, c, d) is a general route which is neither a pseudo-cycle nor a path,
- (a, b, c, d) is a directed path,

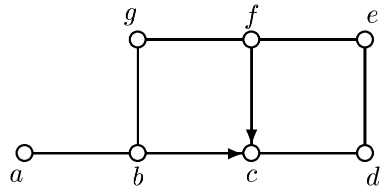


Figure 2: Examples of a routes in a graph.

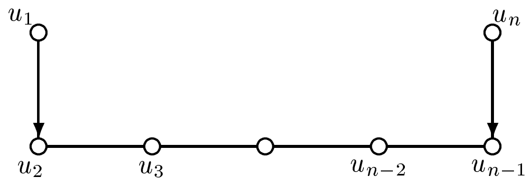


Figure 3: A complex.

- $(b, c, f, e, d, c, f, g, b)$ is a pseudo-cycle which is not a cycle,
- (b, c, d, e, f, g, b) is a directed cycle,
- (a, b, g, f) is both an undirected path and a descending path,
- (d, c, b, a) is a path which is neither undirected nor directed.

A vertex u is an *ancestor* of a vertex v in a graph G if there exists a descending route from u to v in G. Note that every (descending) route ρ can be shortened to a (descending) path. Indeed, if a vertex w occurs more than once in $\rho : (u = u_1, \ldots, u_n = v)$, then ρ can be replaced by $(u_1, \ldots, u_{i-1}, u_k, \ldots, u_n)$ where u_i is the first occurrence of a node w in ρ, and u_k is the last occurrence of w in ρ. The set of ancestors of vertices of a set $U \subseteq V$ is denoted by $\mathrm{an}(U)$.
A *complex* in a hybrid graph G is a path $(u_1, \ldots, u_n)$ in G such that $n > 2$, $u_1 \to u_2$, $u_{n-1} \leftarrow u_n$, $u_i - u_{i+1}$ for all $i = 2, \ldots, n-2$, and no other pair of vertices of $\{u_1, \ldots, u_n\}$ is an edge in G. That means, the subgraph of G induced by $\{u_1, \ldots, u_n\}$ looks like the graph in Figure 3. Note that our concept of complex corresponds to the concept of 'minimal complex' from [8]. An arrow $x \to y$ is called a *complex arrow* in G if there exists a complex $(u_1, \ldots, u_n)$ in G such that $x = u_1$ and $y = u_2$. An arrow $x \to y$ in G is called a *non-complex* arrow if it is not a complex arrow in G. Two graphs will be called *(graph) equivalent*, if they have the same underlying graph and the same complexes. It is evidently an equivalence relation. The following lemma simplifies the task to verify whether two graphs are equivalent.

Lemma 2.1 Two hybrid graphs are graph equivalent iff they have the same underlying graph and the same complex arrows.

Proof: It suffices to show that whenever G_1 and G_2 have the same underlying graph and complex arrows, then they have the same complexes. Suppose for a contradiction that $(u_1, \ldots, u_n)$ is a complex in G_1, which is not a complex in G_2. Since $u_1 \to u_2$ and $u_n \to u_{n-1}$ are complex arrows in G_1, they are arrows in G_2. Let us put $i = \max\{k;\ 1 \le k \le n-2, u_k \to u_{k+1} \text{ in } G_2\}$ and $j = \min\{k;\ i+1 \le k \le n-1, u_k \leftarrow u_{k+1} \text{ in } G_2\}$. Then $(u_{i+1}, \ldots, u_j)$ is an undirected path in G_2. Since $(u_1, \ldots, u_n)$ is a complex in G_1, and G_2 has the same underlying graph as G_1, the path $(u_i, \ldots, u_{j+1})$ is a complex in G_2. One has $i = 1$ as otherwise (u_i, u_{i+1}) is a complex arrow in G_2 which is a line in G_1. Analogously, $j = n-1$ as otherwise (u_{j+1}, u_j) is a complex arrow in G_2 which is a line in G_1. Thus $(u_1, \ldots, u_n)$ is a complex in G_2 which contradicts the assumption. ■

Lemma 2.2 Let $u \to v$ be a non-complex arrow in a hybrid graph G, and the graph H differs from G only in the edge (u, v), which is a line in H. Then G and H are graph equivalent.

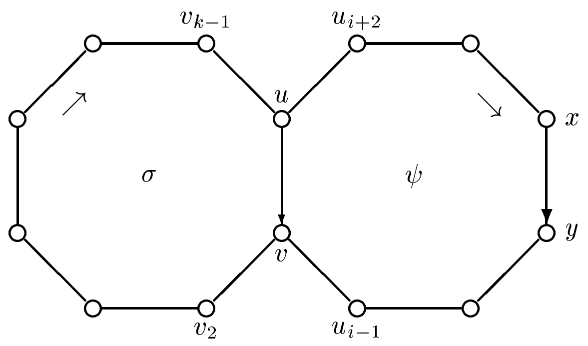


Figure 4: Figure illustrating the proof of Lemma 2.4.

Proof: The graphs G and H have the same underlying graph. By Lemma 2.1 it suffices to verify that they have the same complex arrows. Since $u \to v$ is not a part of any complex in G every complex in G remains a complex in H and every complex arrow in G is a complex arrow in H.

Let us prove by contradiction that every complex arrow in H is a complex arrow in G. Consider a complex arrow $a \to b$ in H which is a non-complex arrow in G. Then there exists a complex $(a, b = c_1, \ldots, c_n, d)$, $n \geq 1$ in H. Since it is not a complex in G the edge (u, v) belongs to the path $(c_1, \ldots, c_n)$. Find the index i such that $u = c_i$ and either $v = c_{i+1}$ or $v = c_{i-1}$. Then either the path $(c_i = u, c_{i+1} = v, \ldots, c_n, d)$ or the path $(a, b = c_1, \ldots, c_{i-1} = v, c_i = u)$ is a complex in G which contradicts the premise that $u \to v$ is a non-complex arrow in G. ■

Consequence 2.3 Let G be a hybrid graph, $\mathcal{A}$ a collection of non-complex arrows in G, and H a graph made of G by converting the arrows from $\mathcal{A}$ into lines. Then H is graph equivalent to G.

Proof: Let us order the collection $\mathcal{A}$ into a sequence $u_i \to v_i$, $i = 1, \ldots, m$ and denote $\mathcal{A}_i = \{u_j \to v_j; i \leq j \leq m\}$ for $i = 1, \ldots, m$. Put $G_1 \equiv G$ and introduce G_{i+1} (for $i = 1, \ldots, m$) as the graph made of G_i by converting the arrow $u_i \to v_i$ into a line. The idea is to show by induction on $i = 1, \ldots, m$ that $\mathcal{A}_i$ is a collection of non-complex arrows in G_i and that G_{i+1} is equivalent to G_i. Indeed, one can apply Lemma 2.2 to show that G_2 is equivalent to G_1. The induction step (for $i = 1, \ldots, m$): since G_{i+1} and G_i have the same complex arrows (Lemma 2.1) $\mathcal{A}_{i+1}$ is a collection of non-complex arrows in G_{i+1} as well. This allows to apply Lemma 2.2 again to show that G_{i+2} is equivalent to G_{i+1}. Hence, $H = G_{m+1}$ is equivalent to $G_1 = G$. ■

Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are hybrid graphs. We will say that G_1 is *larger* than G_2, and write $G_1 \geq G_2$ if $E_1 \supseteq E_2$. It implies that every line in G_2 is a line in G_1. In a particular case that G_1 and G_2 have the same underlying graph, $G_1 \geq G_2$ iff every arrow in G_1 is an arrow in G_2. Note that whenever a vertex u is an ancestor of a vertex v in G_2 and $G_1 \geq G_2$ then u is an ancestor of v in G_1. Indeed, it suffices to realize that a sequence of vertices $(u = u_1, \ldots, u_n = v)$, $n \geq 1$ is a descending route in $G_j = (V, E_j)$, $j = 1, 2$ iff $(u_i, u_{i+1}) \in E_j$ for all $i = 1, \ldots, n-1$.

2.2 Cyclic arrows

Let G be a hybrid graph and $u \to v$ an arrow in G. We will say that $u \to v$ is a *cyclic arrow* in G, if there exists a directed pseudo-cycle in G such that $u \to v$ is part of it. Equivalently, if $u \to v$ in G and v is an ancestor of u in G. In particular, $u \to v$ a cyclic arrow in G iff there exists a directed cycle in G containing $u \to v$.

Lemma 2.4 Let G be a hybrid graph and $u \to v$ a cyclic arrow in G. Let the graph H is made of G by converting $u \to v$ into a line. Then an arrow $x \to y$ is a cyclic arrow in H iff it is a cyclic arrow in G, different from $u \to v$.

Proof: Every directed pseudo-cycle in G containing an arrow $x \to y$ different from $u \to v$ remains a directed pseudo-cycle in H. Thus, every cyclic arrow in G different from $u \to v$ is a cyclic arrow in H.

Conversely, suppose that $x \to y$ is a cyclic arrow in H. Then $x \to y$ in G and there exists a descending route $\psi : (y = u_1, \ldots, u_n = x)$, $n \geq 3$ in H. If ψ remains a descending route in G, then $x \to y$ is a cyclic arrow in G. Otherwise there exists $1 \leq i \leq n-1$ such that $(u_i, u_{i+1}) = (v, u)$. Since $u \to v$ is a cyclic arrow in G there exists a descending route $\sigma : (v = v_1, \ldots, v_k = u)$, $k \geq 3$ in G (see Figure 4). Therefore $(y = u_1, \ldots, u_i = v = v_1, \ldots, v_k = u = u_{i+1}, \ldots, u_n = x)$ is a descending route in G and $x \to y$ is a cyclic arrow in G as well. ■

2.3 Chain graphs

A *chain graph* is a hybrid graph in which there is no directed pseudo-cycle. Equivalently, a chain graph is a hybrid graph without cyclic arrows. In particular, a hybrid graph is a chain graph iff it has no directed cycle. Every undirected graph is a chain graph because it does not contain any arrow. Directed chain graphs are more often called directed acyclic graphs. Note that the above definition of a chain graph is not the original one given by Lauritzen and Wermuth [10] which motivated the name 'chain'. Other equivalent definitions of a chain graph are given in [17], Lemma 2.1. A simple way of how to convert a hybrid graph into a chain graph is based on Lemma 2.4.

Consequence 2.5 Let K be a hybrid graph and H is a graph made of K by converting all its cyclic arrows into lines. Then H is a chain graph.

Proof: Let us order the collection of all cyclic arrows in K into a sequence $u_i \to v_i$, $i = 1, \ldots, m$ and denote $\mathcal{A}_i = \{u_j \to v_j ; i \leq j \leq m\}$ for $i = 1, \ldots, m+1$. Put $G_1 \equiv K$ and introduce G_{i+1} (for $i = 1, \ldots, m$) as the graph made of G_i by converting the arrow $u_i \to v_i$ into a line. One can use Lemma 2.4 to show by induction on $i = 1, \ldots, m+1$ that $\mathcal{A}_i$ is the collection of all cyclic arrows in G_i. Hence, $H = G_{m+1}$ has no cyclic arrow. ■

The graph equivalence decomposes the class of chain graphs over V into equivalence classes. The proof of the following important result was given by Frydenberg [8], Proposition 5.7.

Theorem 2.6 Every equivalence class of graph equivalent chain graphs contains a graph which is larger than any other graph of the class.

Of course, the graph from the previous theorem is uniquely determined. It will be called the *largest chain graph* of the class of equivalent chain graphs. Let us emphasize that the equivalence class may contain incomparable chain graphs (with respect to the relation 'larger') in general. On the other hand, the largest chain graph of the class *is* comparable with every chain graph of the class. The only difference between a general chain graph G belonging to the class and the largest chain graph L of the class is that some non-complex arrows in G can be lines in L.

2.4 Independency models and Markov properties

Let V be a non-empty finite set of variables. Let us denote the set of all triplets $\langle X, Y | Z \rangle$ by $\mathcal{T}(V)$, where X, Y, Z are disjoint subsets of V, and X, Y are non-empty. If the sets X, Y have only one element, then the triplet $\langle X, Y | Z \rangle$ is called *elementary*. The set of all elementary triplets over V is denoted $\mathcal{E}(V)$. An *independency model* over V is any subset of $\mathcal{T}(V)$. An independency model $\mathcal{M}$ is a *semi-graphoid* [14] if it satisfies the following properties:

$$\langle X, Y | Z \rangle \in \mathcal{M} \quad \Leftrightarrow \quad \langle Y, X | Z \rangle \in \mathcal{M}$$
$$\{ \langle X, Y | WZ \rangle \in \mathcal{M} \ \& \ \langle X, W | Z \rangle \in \mathcal{M} \} \quad \Leftrightarrow \quad \langle X, YW | Z \rangle \in \mathcal{M}$$

The significance of elementary triplets is that the list of elementary triplets belonging to a semi-graphoid $\mathcal{M}$ suffices to reconstruct $\mathcal{M}$ and can be used as an economical record of $\mathcal{M}$. We leave it to the reader to verify the following lemma.

Lemma 2.7 Let $\mathcal{M}$ be a semi-graphoid over V, $\langle X, Y | Z \rangle \in \mathcal{T}(V)$. Then $\langle X, Y | Z \rangle \in \mathcal{M}$ iff

$$\forall\, x \in X \quad \forall\, y \in Y \quad \forall\, Z \subseteq W \subseteq (X \cup Y \cup Z) \setminus \{x, y\} \quad \langle x, y | W \rangle \in \mathcal{M} \cap \mathcal{E}(V) \,.$$

In particular, $\mathcal{M}_1 = \mathcal{M}_2$ iff $\mathcal{M}_1 \cap \mathcal{E}(V) = \mathcal{M}_2 \cap \mathcal{E}(V)$ for semi-graphoids $\mathcal{M}_1$, $\mathcal{M}_2$ over V.

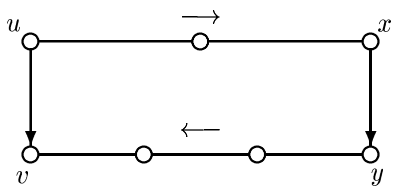


Figure 5: $u \to v$ covers $x \to y$.

Every chain graph over V induces a certain independency model over V. The *moral graph* of a hybrid graph K is an undirected graph over the same set of vertices which has an edge (u, v) iff either (u, v) is an edge in K or there exists a complex $(u = u_1, \ldots, u_n = v)$, $n \geq 3$ in K. We will say that a triplet $\langle X, Y|Z \rangle \in \mathcal{T}(V)$ is *represented* in a chain graph G over V according to the *moralization criterion* if every path in the moral graph of $G_{\mathrm{an}(X \cup Y \cup Z)}$ from a vertex of X to a vertex of Y contains a vertex of Z. The independency model *induced* by G consists of the triplets represented in G according to the moralization criterion. It is always a semi-graphoid - see [18], Lemma 3.1. Thus, according to Lemma 2.7 one can encode it by means of the list of elementary triplets represented in the graph.
Let $\{\mathbf{X}_i;\ i \in V\}$ be a collection of finite non-empty sets indexed by a finite non-empty set V. Let the symbol $\Pi(U)$, where $\emptyset \neq U \subseteq V$, denote the Cartesian product $\prod_{i \in U} \mathbf{X}_i$. A discrete *probability distribution* over V is a function $P : \Pi(V) \to [0, 1]$, which satisfies $\sum_{\mathbf{x} \in \Pi(V)} P(\mathbf{x}) = 1$. The *marginal distribution* of P *for* a non-empty subset $U \subseteq V$ is a probability distribution P^U over U defined by:

$$P^U(\mathbf{x}) = \sum_{\mathbf{y} \in \Pi(V \setminus U)} P(\mathbf{x}, \mathbf{y})$$

for every $\mathbf{x} \in \Pi(U)$. Of course, $P^{\emptyset} \equiv 1$. Supposing $\langle X, Y|Z \rangle \in \mathcal{T}(V)$ we say that X is *conditionally independent of Y given Z with respect to P* if

$$\forall\ \mathbf{x} \in \Pi(X) \quad \mathbf{y} \in \Pi(Y) \quad \mathbf{z} \in \Pi(Z) \quad P^{X \cup Y \cup Z}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \cdot P^Z(\mathbf{z}) = P^{X \cup Z}(\mathbf{x}, \mathbf{z}) \cdot P^{Y \cup Z}(\mathbf{y}, \mathbf{z})\,.$$

The independency model *induced* by a probability distribution P consists of the triplets $\langle X, Y|Z \rangle \in \mathcal{T}(V)$ such that X is conditionally independent of Y given Z with respect to P. Note that it is always a semi-graphoid as well [7].
A probability distribution P over V is called *Markovian* with respect to a chain graph G over V if the independency model induced by G is a subset of the independency model induced by P. Two chain graphs over the same set of nodes are *Markov equivalent* if their classes of Markovian distributions coincide. Frydenberg [8] gave the following elegant characterization of Markov equivalent chain graphs. One can use it to show that two chain graphs are Markov equivalent iff they induce the same independency model - see also [1].

Theorem 2.8 Two chain graphs are Markov equivalent iff they are graph equivalent.

3 Characterization of the largest chain graphs

3.1 Protected arrows

The goal of this section is to characterize arrows in the largest chain graph of a class of equivalent chain graphs. It seems very easy – an edge is an arrow in the largest chain graph iff it is an arrow in every equivalent chain graph. However, to inspect the whole equivalence class of chain graphs is rather demanding. Thus, a reasonable characterization of arrows in the largest chain graph should work only with one graph from the equivalence class. We have found out that every non-complex arrow in the largest chain graph prevents a complex arrow from being a cyclic arrow.

Definition 3.1 Let G be a hybrid graph. We say that an arrow $u \to v$ in G *covers* an arrow $x \to y$ in G and write $u \to v \succeq x \to y$ if u is an ancestor of x in G and y is an ancestor of v in G

(see Figure 5). We say that an arrow $u \to v$ is *protected* in G if it covers a complex arrow in G. An arrow in G is called *non-protected* if it is not a protected arrow in G.

Since every vertex is an ancestor of itself the relation $\succeq$ is reflexive. Thus, every complex arrow is a protected arrow. Since the relation 'being an ancestor' is transitive the relation $\succeq$ is transitive as well. In particular, an arrow which covers a protected arrow is a protected arrow.

Lemma 3.1 Let G be a chain graph, and $u \to v$ and $x \to y$ in G. Then $u \to v \succeq x \to y$ in G iff there exists a descending path from u to v containing the arrow $x \to y$ in G.

Proof: The sufficiency of the given condition is trivial. For necessity suppose $u \to v \succeq x \to y$. Let $(u = u_1, \ldots, u_n = x)$, $n \geq 1$ and $(y = v_1, \ldots, v_m = v)$, $m \geq 1$ be the corresponding descending paths. We prove by contradiction that $\{u_1, \ldots, u_n\} \cap \{v_1, \ldots, v_m\} = \emptyset$. Let $k \in \{1, \ldots, n\}$ be the largest index, for which there exists an index $i \in \{1, \ldots, m\}$ such that $u_k = v_i$. If $k = n$, then $i \neq 1$ since $x \neq y$. Moreover, $i \neq 2$ as otherwise $v_2 = u_n = x \to y = v_1$ contradicts the fact that $[v_1 \to v_2$ or $v_1 — v_2]$. Thus, $k = n$ implies $i \geq 3$. Analogously, $i = 1$ implies $k \leq n - 2$. Then the route $(u_k, \ldots, u_n = x, y = v_1, \ldots, v_i)$ is a directed cycle in G which contradicts the assumption that G is a chain graph. ■

Consequence 3.2 Let G be a chain graph and $u \to v$, $x \to y$ two different arrows in G. Let the graph K differs from G only in that the edge (u, v) is a line in K. Then $u \to v \succeq x \to y$ in G iff the arrow $x \to y$ is a cyclic arrow in K.

Proof: Supposing $u \to v \succeq x \to y$ by Lemma 3.1 there exists a directed path $(u = u_1, \ldots, u_n = x, y = v_1, \ldots, v_m = v)$ in G containing the arrow $x \to y$ where $m + n \geq 3$. Then $(u = u_1, \ldots, u_n = x, y = v_1, \ldots, v_m = v, u)$ is a directed cycle in K.
Conversely, let $x \to y$ be a cyclic arrow in K. Thus, there exists a directed cycle in K containing $x \to y$. Since G does not contain a directed cycle and G and K differ only in the type of (u, v) the above mentioned directed cycle involves (u, v). The sections $(y, \ldots, v)$ and $(u, \ldots, x)$ of the cycle are descending paths both in G and K then. ■

Lemma 3.3 Let G be a chain graph, $u \to v$ an arrow in G, and the graph H is made of G by converting all its arrows, which are covered by $u \to v$ in G (including $u \to v$) into lines. Then H is a chain graph.

Proof: Let us transform G into H in two steps. First, we replace only the arrow $u \to v$ by a line and obtain a hybrid graph K. Consequence 3.2 says that an arrow $x \to y$ is a cyclic arrows in K iff it is covered by $u \to v$ in G but differs from $u \to v$. Second, we convert all cyclic arrows in K into lines and obtain the graph H. By Consequence 2.5 H is a chain graph. ■

3.2 Main results

Lemma 3.4 Let G be a chain graph and L the largest chain graph equivalent to G. Then every non-protected arrow in G is a line in L.

Proof: Let $u \to v$ be a non-protected arrow in G. Let us create the graph H by converting all arrows in G, which are covered by $u \to v$ in G (including $u \to v$) into lines. By Lemma 3.3 H is a chain graph. Every arrow in G covered by $u \to v$ is a non-complex arrow in G (otherwise $u \to v$ is protected in G). Thus, by Consequence 2.3 H is graph equivalent to G. Thus, H is a chain graph equivalent to G, but strictly larger than G because (u, v) is a line in H. Since L is equivalent to H but larger than H, (u, v) is a line in L as well. ■

Lemma 3.5 Let G and H are equivalent chain graphs and $H \geq G$. Then every protected arrow $u \to v$ in G is a protected arrow in H.

Proof: According to the assumption $u \to v$ covers in G a complex arrow $x \to y$. Because G and H are graph equivalent $x \to y$ is also a complex arrow in H. Since u is an ancestor of x in G and $H \geq G$, u is an ancestor of x in H as well. For similar reasons y is an ancestor of v in H. In particular, there exists a directed route in H from u to v containing $x \to y$. Thus the edge (u, v) in H must be an arrow from u to v as otherwise there exists a directed pseudo-cycle in H. Evidently, $u \to v$ covers $x \to y$ in H. ■

Consequence 3.6 Let G be a chain graph and L the largest chain graph equivalent to G. Then $u \to v$ is an arrow in L iff $u \to v$ is a protected arrow in G.

Proof: If $u \to v$ is a protected arrow in G, then $u \to v$ is an arrow in L by Lemma 3.5. The converse follows from Lemma 3.4. ■

Theorem 3.7 A chain graph G is the largest chain graph of the class of all its graph equivalent chain graphs iff every arrow in G is protected in G.

Proof: To show that every arrow in G is protected in G apply Consequence 3.6 with $G = L$. Conversely, suppose for a contradiction that every arrow in G is protected in G but there exists a chain graph $H \neq G$ equivalent to G and larger than G. There exists an edge (u, v), which is an arrow in G and a line in H. According to the assumption $u \to v$ is a protected arrow in G. Lemma 3.5 implies that $u \to v$ is an arrow in H as well, which contradicts the fact that $u — v$ in H. ■

Theorem 3.7 gives an answer to the question whether a given chain graph is the largest chain graph of a class of equivalent chain graphs or not. In case the answer is negative we would like to be able to construct the respective largest chain graph.

Consequence 3.8 The set of protected arrows is the same for all equivalent chain graphs.

Proof: It follows directly from Consequence 3.6. ■

Theorem 3.9 Let G be a chain graph. Let H be the hybrid graph obtained from G by replacing all non-protected arrows in G by lines. Then H is the largest chain graph of the class of chain graphs equivalent to G.

Proof: Let L denote the corresponding largest chain graph. According to Theorem 3.7 an edge (u, v) in L is an arrow $u \to v$ in L iff it is a protected arrow in L. According to Consequence 3.8 an edge (u, v) in L is a protected arrow in L iff it is a protected arrow in G. Since G and L have the same underlying graph the graphs L and H must coincide. ■

Theorem 3.9 can be used as a basis for an evident algorithm constructing the largest chain graph of the class of chain graphs which are equivalent to a given chain graph G:

1. Find and indicate all non-protected arrows in G.
2. Convert all indicated arrows into lines.

One can also consider the following algorithm which is based mainly on lemmas from the preceding subsection.

1. Seek for a non-protected arrow in G. If there is no such arrow in G, then G is the largest chain graph.
2. Convert the chosen non-protected arrow into a line and denote the resulting graph H.
3. Seek for a cyclic arrow in H. If there is no such arrow in H, then put $G \equiv H$ and return to 1.
4. Convert the chosen cyclic arrow into a line and return to 3.

Indeed, if there is no non-protected arrow in the chain graph G in Step 1, then G is the largest chain graph by Theorem 3.7. If there is a non-protected arrow in G, then it is a non-complex arrow and by Lemma 2.2 the graph H in Step 2 is equivalent to G. Repetitive application of Steps 3 and 4 leads to a chain graph by Consequence 2.5. Consequence 2.3 implies that the resulting graph is equivalent to the original graph G. Note for explanation that if one converts in Step 2 a *protected* non-complex arrow into a line, then a complex arrow in G becomes a cyclic arrow in H (see Consequence 3.2). Thus, the resulting graph after Steps 3 and 4 is then a chain graph which is *not* equivalent to the original graph G.

4 Catalog of the largest chain graphs

The goal of this section is to give a catalog of all largest chain graphs over n vertices, $2 \leq n \leq 5$, together with the induced independency models. Since isomorphic graphs need not be repeated just one representative is given for each equivalence class of isomorphic graphs. Every independency models induced by a graph in the catalog is recorded in the form of an encoded list of represented elementary triplets.

4.1 Preliminaries

To help the reader to get a picture we give some numbers below.

Lemma 4.1 The number of all hybrid graphs over n vertices is given by the formula $4^{\binom{n}{2}}$.

Proof: Let us order the set of vertices into a sequence $u_1, \ldots, u_n$. The number of all ordered pairs (u_i, u_j), $i < j$ is then $\binom{n}{2}$. In a hybrid graph, for every such a pair of vertices just one of the following possibilities occurs: line, arrow, reverse arrow or non-edge. ■

Lemma 4.2 The number of all elementary triplets over n variables is $n \cdot (n-1) \cdot 2^{n-2}$. The number of bits needed to encode a semi-graphoid over n variables is $\binom{n}{2} \cdot 2^{n-2}$.

Proof: The number of all ordered pairs of distinct elements of an n-element set is $n \cdot (n-1)$. Supposing we have chosen the first two components of an elementary triplet it remains $n-2$ variables. The number of all subsets of that $(n-2)$-element set is 2^{n-2}. However, to record a semi-graphoid $\mathcal{M}$ in a form of a list of elementary triplets (see Lemma 2.7) one does not need to reserve in memory of a computer bits for all elementary triplets. Since $\langle x, y|W\rangle \in \mathcal{M}$ iff $\langle y, x|W\rangle \in \mathcal{M}$ it suffices to allocate just one bit for such a pair of 'mutually symmetric' triplets. ■

Table 1 gives some numbers of graphs over n vertices, $2 \leq n \leq 5$, which were obtained by a computer program. In the table, LCG means 'largest chain graph', DAG 'directed acyclic graph' and UG 'undirected graph'. Note that we don't know the exact numbers of those graphs for $n \geq 6$, except for chain graphs (28903216) and largest chain graphs (1853976) over 6 vertices.

Table 1: Some numbers.

Number of vertices	2	3	4	5
Number of hybrid graphs	4	64	4096	1048576
Number of chain graphs	4	50	1688	142624
Number of LCGs	2	11	200	11519
Number of LCGs, which are equivalent to a DAG	2	11	185	8782
Number of LCGs, which are equivalent to an UG	2	8	64	1024
Number of LCGs equivalent both to an UG and a DAG	2	8	61	822
Number of LCGs, which are not equivalent to a DAG or an UG	0	0	12	2535
Number of non-isomorphic largest chain graphs	2	5	22	181
Number of bits needed to encode a semi-graphoid	1	6	24	80

From every pair of mutually symmetric triplets over $\{a, b, c, d, e\}$ we choose that one whose first component precedes the second component in the sequence a, b, c, d, e. Table 2 encodes these elementary triplet into numbers. To spare space we refer to a particular elementary triplet by this number in sequel. For example, the number 45 refers to the triplet $\langle d, e|c\rangle$. In the table, ab means $\{a, b\}$.

Table 2: Elementary triplets over $\{a, b, c, d, e\}$.

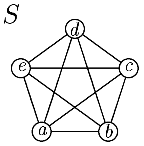
	0	10	20	30	40	50	60	70
0	$\langle a,b\|\emptyset\rangle$	$\langle a,c\|d\rangle$	$\langle a,d\|bc\rangle$	$\langle a,d\|e\rangle$	$\langle c,e\|a\rangle$	$\langle a,d\|be\rangle$	$\langle b,e\|ad\rangle$	$\langle a,b\|cde\rangle$
1	$\langle a,c\|\emptyset\rangle$	$\langle a,d\|b\rangle$	$\langle b,c\|ad\rangle$	$\langle b,c\|e\rangle$	$\langle c,e\|b\rangle$	$\langle a,d\|ce\rangle$	$\langle b,e\|cd\rangle$	$\langle a,c\|bde\rangle$
2	$\langle b,c\|\emptyset\rangle$	$\langle a,d\|c\rangle$	$\langle b,d\|ac\rangle$	$\langle b,d\|e\rangle$	$\langle c,e\|d\rangle$	$\langle a,e\|bc\rangle$	$\langle c,d\|ae\rangle$	$\langle a,d\|bce\rangle$
3	$\langle a,b\|c\rangle$	$\langle b,c\|d\rangle$	$\langle c,d\|ab\rangle$	$\langle c,d\|e\rangle$	$\langle d,e\|a\rangle$	$\langle a,e\|bd\rangle$	$\langle c,d\|be\rangle$	$\langle a,e\|bcd\rangle$
4	$\langle a,c\|b\rangle$	$\langle b,d\|a\rangle$	$\langle a,e\|\emptyset\rangle$	$\langle a,e\|b\rangle$	$\langle d,e\|b\rangle$	$\langle a,e\|cd\rangle$	$\langle c,e\|ab\rangle$	$\langle b,c\|ade\rangle$
5	$\langle b,c\|a\rangle$	$\langle b,d\|c\rangle$	$\langle b,e\|\emptyset\rangle$	$\langle a,e\|c\rangle$	$\langle d,e\|c\rangle$	$\langle b,c\|ae\rangle$	$\langle c,e\|ad\rangle$	$\langle b,d\|ace\rangle$
6	$\langle a,d\|\emptyset\rangle$	$\langle c,d\|a\rangle$	$\langle c,e\|\emptyset\rangle$	$\langle a,e\|d\rangle$	$\langle a,b\|ce\rangle$	$\langle b,c\|de\rangle$	$\langle c,e\|bd\rangle$	$\langle b,e\|acd\rangle$
7	$\langle b,d\|\emptyset\rangle$	$\langle c,d\|b\rangle$	$\langle d,e\|\emptyset\rangle$	$\langle b,e\|a\rangle$	$\langle a,b\|de\rangle$	$\langle b,d\|ae\rangle$	$\langle d,e\|ab\rangle$	$\langle c,d\|abe\rangle$
8	$\langle c,d\|\emptyset\rangle$	$\langle a,b\|cd\rangle$	$\langle a,b\|e\rangle$	$\langle b,e\|c\rangle$	$\langle a,c\|be\rangle$	$\langle b,d\|ce\rangle$	$\langle d,e\|ac\rangle$	$\langle c,e\|abd\rangle$
9	$\langle a,b\|d\rangle$	$\langle a,c\|bd\rangle$	$\langle a,c\|e\rangle$	$\langle b,e\|d\rangle$	$\langle a,c\|de\rangle$	$\langle b,e\|ac\rangle$	$\langle d,e\|bc\rangle$	$\langle d,e\|abc\rangle$

4.2 The catalog

To keep the size of the catalog in reasonable limits and not to loose relevant information the catalog contains only one item for every class of isomorphic graphs.

Figure 6 explains the format of every item of the catalog. It consists of the picture of the largest chain graph, the serial number (S), the number of elements of the class of graph equivalent chain graphs (Q), the number of isomorphic classes (I), the codes of elementary triplets from Table 2 which belong to the corresponding induced independency model (*base*), and the number of elements of this base (M). The symbol in the position T indicates a special property: $T =$ DAG means that the equivalence class contains a directed acyclic graph, $T = *$ means that the class does not contain any directed acyclic or undirected graph, no symbol in the position T means that neither of these two possibilities occurs. Note that the equivalence class contains an undirected graph iff the picture does not contain an arrow.

S



Q	I	M	T
	base		

Figure 6: Format of items of the catalog.

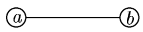
Thus, for a given chain graph G there exists $I \cdot Q$ chain graphs, which are equivalent to a graph isomorphic to G.

4.2.1 Catalog of LCGs over two vertices

1

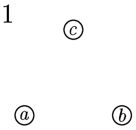
1	1	1	DAG
	0		

2

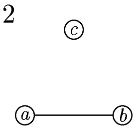


3	1	0	DAG

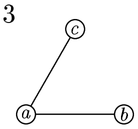
4.2.2 Catalog of LCGs over three vertices

1


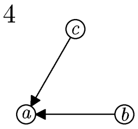
1	1	6	DAG
0–5			

2


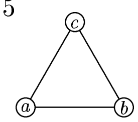
3	3	4	DAG
1, 2, 4, 5			

3


8	3	1	DAG
5			

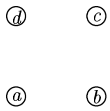
4


1	3	1	DAG
2			

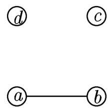
5


13	1	0	DAG

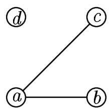
4.2.3 Catalog of LCGs over four vertices

1


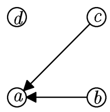
1	1	24	DAG
0–23			

2


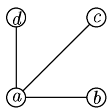
3	6	20	DAG
1, 2, 4–8, 10–17, 19–23			

3


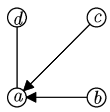
8	12	14	DAG
5–8, 11, 12, 14–17, 20–23			

4


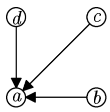
1	12	14	DAG
2, 6–8, 11–17, 20, 22, 23			

5


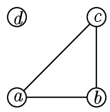
20	4	6	DAG
5, 14, 16, 21–23			

6


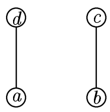
2	12	5	DAG
2, 14, 16, 22, 23			

7


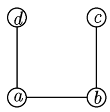
1	4	6	DAG
2, 7, 8, 13, 15, 17			

8


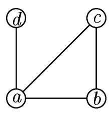
13	4	12	DAG
6–8, 11, 12, 14–17, 20, 22, 23			

9


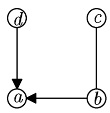
9	3	16	DAG
0, 1, 3, 4, 7–10, 14–19, 22, 23			

10


20	12	7	DAG
4, 14, 16, 17, 19, 22, 23			

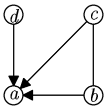
11


32	12	4	DAG
14, 16, 22, 23			

12


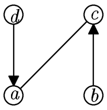
3	24	7	DAG
4, 7, 8, 15, 17, 19, 23			

13



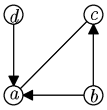
3	12	4	DAG
7, 8, 15, 17			

14



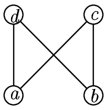
1	12	3	*
7, 18, 23			

15



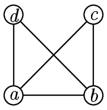
2	24	2	DAG
7, 23			

16



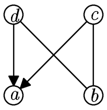
17	3	2	
18, 23			

17



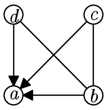
40	6	1	DAG
23			

18



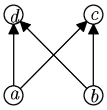
8	12	2	DAG
17, 18			

19



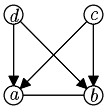
8	12	1	DAG
17			

20



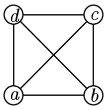
1	6	2	DAG
0, 23			

21



3	6	1	DAG
8			

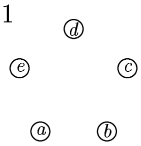
22



75	1	0	DAG

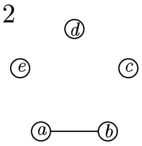
4.2.4 Catalog of LCGs over five vertices

1



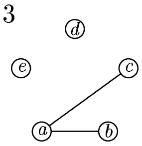
1	1	80	DAG
0–79			

2



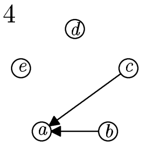
3	10	72	DAG
1, 2, 4–8, 10–17, 19–27, 29–45, 48–69, 71–79			

3



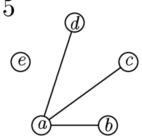
8	30	60	DAG
5–8, 11, 12, 14–17, 20–27, 30, 32–45, 50–55, 57–69, 72–79			

4



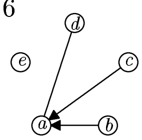
1	30	60	DAG
2, 6–8, 11–17, 20, 22–27, 30–45, 50–54, 56–69, 72, 73, 75–79			

5



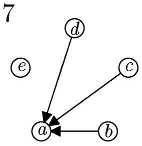
20	20	44	DAG
5, 14, 16, 21–27, 34–45, 52–55, 57, 59–62, 64–69, 73–79			

6



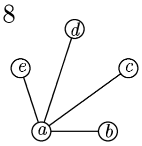
2	60	42	DAG
2, 14, 16, 22–27, 31, 34–45, 52–54, 57, 59–62, 64–69, 73, 75–79			

7



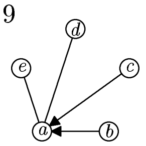
1	20	44	DAG
2, 7, 8, 13, 15, 17, 24–27, 31–45, 52–54, 56, 58–61, 63–69, 73, 76, 78, 79			

8



48	5	24	DAG
5, 14, 16, 21–23, 37, 40, 43, 55, 57, 59, 60, 62, 64, 65, 67, 68, 74–79			

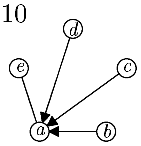
9



4	30	21	DAG

2, 14, 16, 22, 23, 37, 40, 43, 57, 59, 60, 62, 64, 65, 67, 68, 75–79

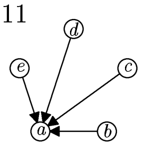
10



2	20	18	DAG

2, 7, 8, 13, 15, 17, 37, 40, 43, 59, 60, 64, 65, 67, 68, 76, 78, 79

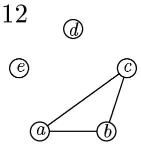
11



1	5	24	DAG

2, 7, 8, 13, 15, 17, 25–27, 31–33, 38, 39, 41, 42, 44, 45, 56, 58, 61, 63, 66, 69

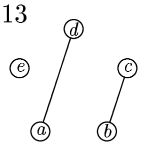
12



13	10	56	DAG

6–8, 11, 12, 14–17, 20, 22–27, 30, 32–45, 50–54, 57–69, 72, 73, 75–79

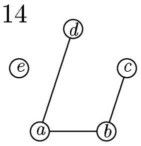
13



9	15	64	DAG

0, 1, 3, 4, 7–10, 14–19, 22–29, 32–49, 52–54, 57–71, 73, 75–79

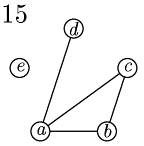
14



20	60	46	DAG

4, 14, 16, 17, 19, 22–27, 34–45, 48, 52–54, 57, 59–69, 71, 73, 75–79

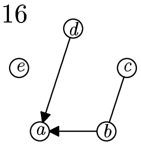
15



32	60	40	DAG

14, 16, 22–27, 34–45, 52–54, 57, 59–62, 64–69, 73, 75–79

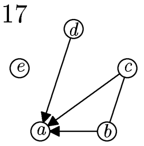
16



3	120	46	DAG

4, 7, 8, 15, 17, 19, 23–27, 32–45, 48, 52–54, 58–61, 63–69, 71, 73, 76–79

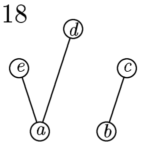
17



3	60	40	DAG

7, 8, 15, 17, 24–27, 32–45, 52–54, 58–61, 63–69, 73, 76, 78, 79

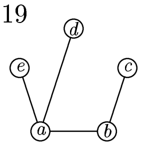
18



24	30	52	DAG

0, 1, 3, 4, 7–10, 14–19, 22, 23, 25, 26, 28, 29, 32, 33, 37–43, 46–49, 57–68, 70, 71, 75–79

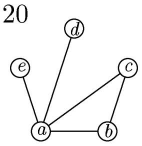
19



48	60	28	DAG

4, 14, 16, 17, 19, 22, 23, 37, 40, 41, 43, 48, 57, 59, 60, 62–68, 71, 75–79

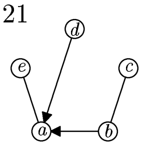
20



76	30	20	DAG

14, 16, 22, 23, 37, 40, 43, 57, 59, 60, 62, 64, 65, 67, 68, 75–79

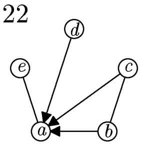
21



6	120	25	DAG

4, 7, 8, 15, 17, 19, 23, 37, 40, 41, 43, 48, 59, 60, 63–68, 71, 76–79

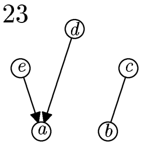
22



6	60	16	DAG

7, 8, 15, 17, 37, 40, 43, 59, 60, 64, 65, 67, 68, 76, 78, 79

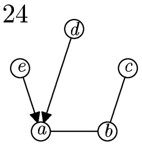
23



3	30	52	DAG

0, 1, 3, 4, 7–10, 14–19, 22, 23, 25–29, 32, 33, 37–42, 44–49, 57–66, 69–71, 75–78

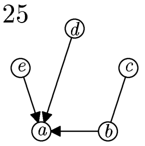
24



4	60	25	DAG

4, 14, 16, 17, 19, 22, 23, 27, 37, 40, 41, 48, 57, 59, 60, 62–66, 71, 75–78

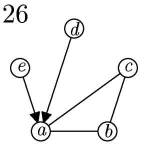
25



3	60	28	DAG

4, 7, 8, 15, 17, 19, 23, 25–27, 32, 33, 38, 39, 41, 42, 44, 45, 48, 58, 61, 63, 64, 66, 69, 71, 77, 78

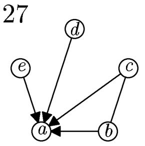
26



6	30	17	DAG

14, 16, 22, 23, 27, 37, 40, 57, 59, 60, 62, 64, 65, 75–78

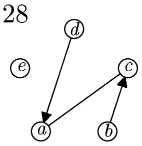
27



3	30	20	DAG

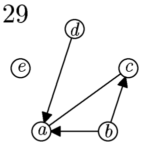
7, 8, 15, 17, 25–27, 32, 33, 38, 39, 41, 42, 44, 45, 58, 61, 63, 66, 69

28



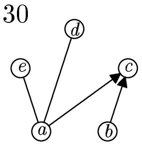
1	60	38	*

7, 18, 23–27, 32, 34–45, 52–54, 59–61, 64–70, 73, 76–79

29 

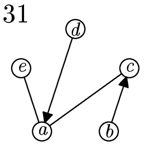
2	120	36	DAG

7, 23–27, 32, 34–45, 52–54, 59–61, 64–69, 73, 76–79

30 

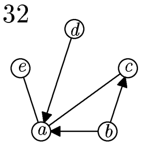
8	60	28	DAG

0, 7, 9, 14, 16, 22, 23, 25, 28, 32, 37, 39, 40, 43, 47, 57, 59, 60, 62, 64, 65, 67, 68, 75–79

31 

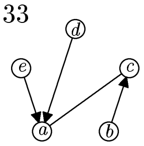
2	120	18	*

7, 18, 23, 37, 40, 43, 59–61, 64, 65, 67, 68, 70, 76–79

32 

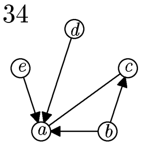
4	120	15	DAG

7, 23, 37, 40, 43, 59, 60, 64, 65, 67, 68, 76–79

33 

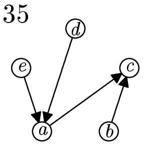
1	60	11	*

7, 23, 25, 27, 32, 39, 44, 64, 70, 77, 78

34 

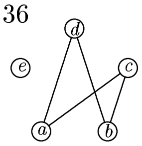
2	60	10	DAG

7, 23, 25, 27, 32, 39, 44, 64, 77, 78

35 

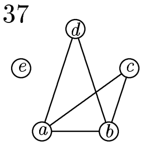
1	60	26	DAG

0, 7, 9, 14, 16, 22, 23, 25, 27, 28, 32, 37, 39, 40, 44, 47, 57, 59, 60, 62, 64, 65, 75–78

36 

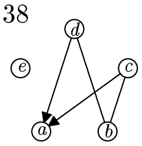
17	15	36	

18, 23–27, 34–45, 52–54, 59–61, 64–70, 73, 76–79

37 

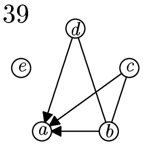
40	30	34	DAG

23–27, 34–45, 52–54, 59–61, 64–69, 73, 76–79

38 

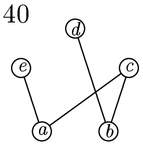
8	60	36	DAG

17, 18, 24–27, 34–45, 52–54, 59–61, 63–70, 73, 76, 78, 79

39 

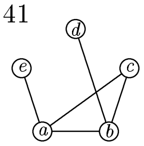
8	60	34	DAG

17, 24–27, 34–45, 52–54, 59–61, 63–69, 73, 76, 78, 79

40 

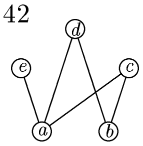
48	60	31	DAG

3, 11, 12, 17, 18, 20, 23, 37, 38, 40, 43–46, 50, 51, 59–61, 63–65, 67–70, 72, 76–79

41 

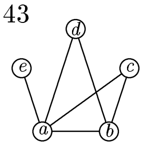
76	60	22	DAG

11, 17, 20, 23, 37, 40, 43, 44, 50, 59, 60, 63–65, 67–69, 72, 76–79

42 

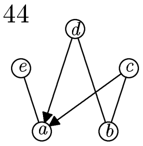
40	60	17	

18, 23, 37, 40, 43, 59–61, 64, 65, 67, 68, 70, 76–79

43 

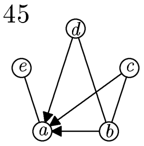
96	60	14	DAG

23, 37, 40, 43, 59, 60, 64, 65, 67, 68, 76–79

44 

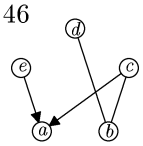
16	60	16	DAG

17, 18, 37, 40, 43, 59–61, 64, 65, 67, 68, 70, 76, 78, 79

45 

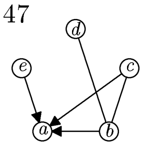
16	60	13	DAG

17, 37, 40, 43, 59, 60, 64, 65, 67, 68, 76, 78, 79

46 

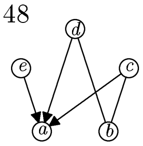
8	120	31	DAG

3, 11, 12, 17, 18, 20, 23, 25–27, 38, 39, 41, 42, 44–46, 50, 51, 59, 61, 63, 66–70, 72, 76, 77, 79

47 

8	120	22	DAG

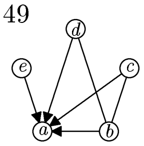
11, 17, 20, 23, 25–27, 38, 39, 41, 42, 44, 45, 50, 61, 63, 66, 67, 69, 72, 77, 79

48 

8	60	17	DAG

17, 18, 25–27, 38, 39, 41, 42, 44, 45, 61, 63, 66, 69, 70, 76

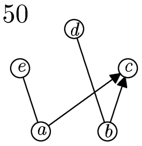
49



8	60	14	DAG

17, 25–27, 38, 39, 41, 42, 44, 45, 61, 63, 66, 69

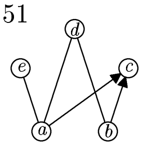
50



9	60	31	DAG

0, 6, 9, 11, 17, 20, 23, 25, 27, 28, 30, 37, 39, 40, 43, 44, 47, 50, 59, 60, 63–65, 67–69, 72, 76–79

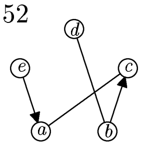
51



20	120	17	DAG

9, 23, 37, 39, 40, 43, 47, 59, 60, 64, 65, 67, 68, 76–79

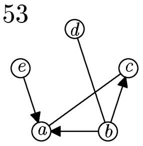
52



3	120	20	*

11, 17, 20, 23, 25, 27, 39, 44, 46, 50, 51, 63, 64, 67, 69, 70, 72, 77–79

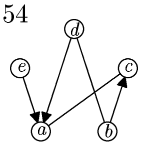
53



6	120	17	DAG

11, 17, 20, 23, 25, 27, 39, 44, 50, 63, 64, 67, 69, 72, 77–79

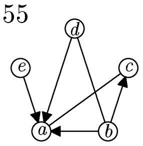
54



3	120	9	*

23, 25, 27, 39, 44, 64, 70, 77, 78

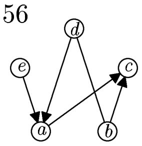
55



6	120	8	DAG

23, 25, 27, 39, 44, 64, 77, 78

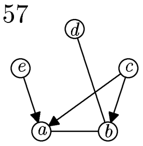
56



3	120	13	DAG

9, 23, 25, 27, 39, 44, 47, 60, 64, 65, 76–78

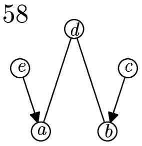
57



4	120	16	DAG

11, 17, 20, 23, 26, 44, 50, 59, 63, 67–69, 72, 76, 77, 79

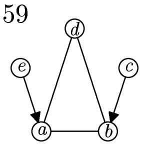
58



1	60	16	*

18, 23, 26, 47–49, 59, 61, 63, 67, 68, 70, 71, 76, 77, 79

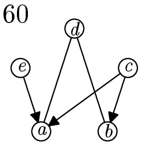
59



2	60	11	*

23, 26, 48, 59, 63, 67, 68, 71, 76, 77, 79

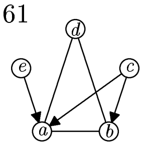
60



2	120	11	*

18, 23, 26, 59, 61, 67, 68, 70, 76, 77, 79

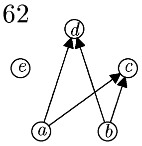
61



4	120	8	DAG

23, 26, 59, 67, 68, 76, 77, 79

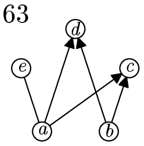
62



1	30	36	DAG

0, 23–28, 34–45, 52–54, 59–61, 64–69, 73, 76–79

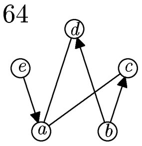
63



3	60	17	DAG

0, 23, 25, 28, 37, 40, 43, 59, 60, 64, 65, 67, 68, 76–79

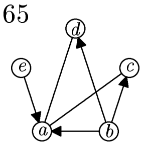
64



1	60	8	*

23, 25, 64, 67, 70, 77–79

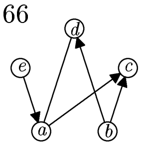
65



4	60	7	DAG

23, 25, 64, 67, 77–79

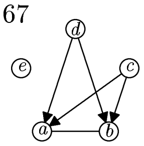
66



1	120	8	*

23, 25, 47, 64, 67, 77–79

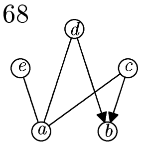
67



3	30	34	DAG

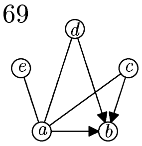
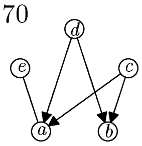
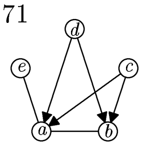
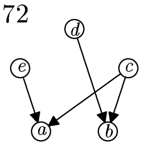
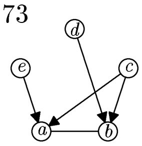
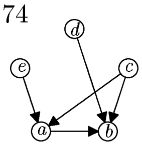
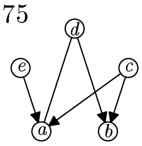
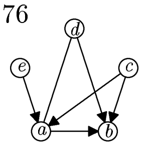
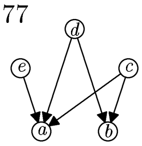
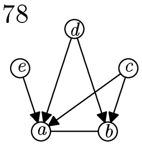
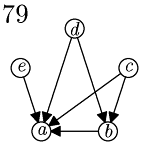
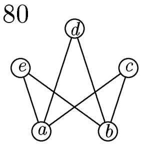
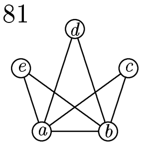
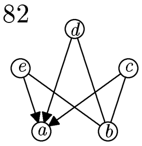
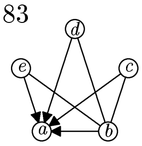
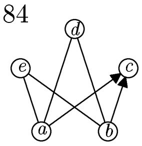
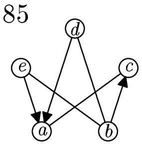
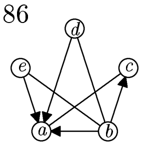
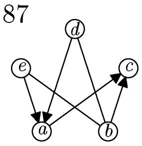
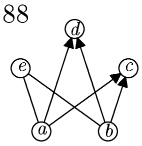
8, 24–27, 33–45, 52–54, 59–61, 64–69, 73, 76, 78, 79

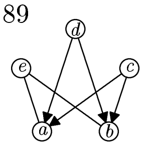
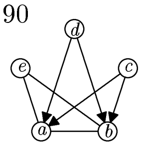
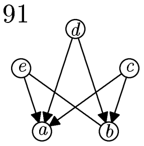
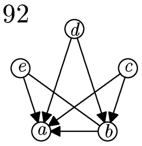
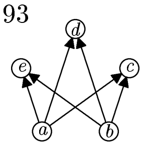
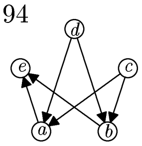
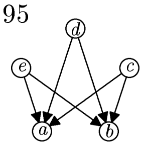
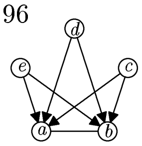
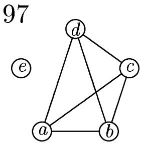
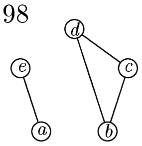
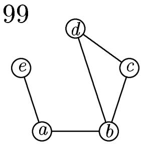
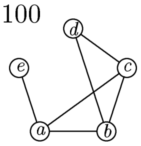
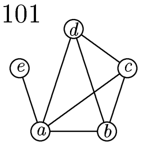
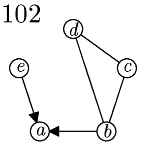
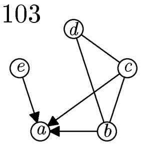
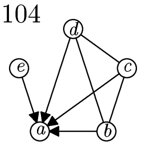
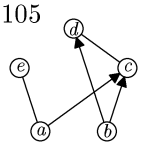
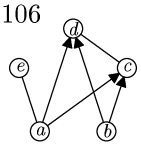
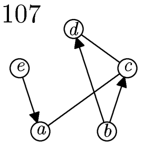
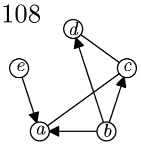
68



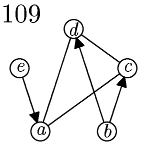
20	60	17	DAG

16, 18, 37, 40, 43, 59–62, 64, 65, 67, 68, 70, 76, 78, 79

No.	Graph					Entries
69		20	60	14	DAG	16, 37, 40, 43, 59, 60, 62, 64, 65, 67, 68, 76, 78, 79
70		2	60	16	DAG	8, 18, 37, 40, 43, 59–61, 64, 65, 67, 68, 70, 76, 78, 79
71		6	60	13	DAG	8, 37, 40, 43, 59, 60, 64, 65, 67, 68, 76, 78, 79
72		1	60	31	DAG	3, 6, 8, 12, 16, 18, 20, 25–27, 30, 33, 38, 39, 41–46, 51, 59, 61, 62, 66, 68–70, 72, 76, 79
73		1	60	8	*	8, 26, 27, 33, 42, 45, 72, 76
74		1	120	17	DAG	6, 8, 12, 16, 26, 27, 30, 33, 42, 43, 45, 51, 59, 62, 68, 76, 79
75		2	120	11	DAG	16, 18, 26, 43, 59, 61, 62, 68, 70, 76, 79
76		2	120	8	DAG	16, 26, 43, 59, 62, 68, 76, 79
77		1	60	17	DAG	8, 18, 25–27, 33, 38, 39, 41, 42, 44, 45, 61, 66, 69, 70, 76
78		2	60	7	DAG	8, 26, 27, 33, 42, 45, 76
79		1	60	14	DAG	8, 25–27, 33, 38, 39, 41, 42, 44, 45, 61, 66, 69
80		24	10	7		23, 64, 67, 70, 77–79
81		108	10	6	DAG	23, 64, 67, 77–79
82		20	20	7	DAG	17, 41, 44, 63, 66, 69, 70
83		20	20	6	DAG	17, 41, 44, 63, 66, 69
84		17	30	7	*	23, 47, 64, 67, 77–79
85		8	60	6	*	23, 44, 64, 70, 77, 78
86		16	60	5	DAG	23, 44, 64, 77, 78
87		8	60	6	DAG	23, 44, 47, 64, 77, 78
88		8	30	7	DAG	23, 28, 64, 67, 77–79

89		3	30	6	*	8, 64, 67, 70, 78, 79
90		6	30	5	DAG	8, 64, 67, 78, 79
91		2	60	6	DAG	8, 41, 44, 66, 69, 70
92		2	60	5	DAG	8, 41, 44, 66, 69
93		1	10	7	DAG	0, 23, 64, 67, 77–79
94		1	30	6	DAG	8, 18, 64, 67, 78, 79
95		1	10	7	DAG	8, 26, 27, 33, 42, 45, 70
96		3	10	6	DAG	8, 26, 27, 33, 42, 45
97		75	5	32	DAG	24–27, 34–45, 52–54, 59–61, 64–69, 73, 76, 78, 79
98		39	10	48	DAG	0, 1, 3, 4, 6, 9–12, 18–20, 25–30, 37–51, 59–61, 64–72, 76, 78, 79
99		76	60	24	DAG	4, 11, 19, 20, 37, 40, 41, 43, 44, 48, 50, 59, 60, 64–69, 71, 72, 76, 78, 79
100		92	60	15	DAG	20, 37, 40, 43, 59, 60, 64, 65, 67–69, 72, 76, 78, 79
101		176	20	12	DAG	37, 40, 43, 59, 60, 64, 65, 67, 68, 76, 78, 79
102		13	60	24	DAG	4, 11, 19, 20, 25–27, 38, 39, 41, 42, 44, 45, 48, 50, 61, 64, 66, 67, 69, 71, 72, 78, 79
103		13	60	15	DAG	20, 25–27, 38, 39, 41, 42, 44, 45, 61, 66, 69, 72, 79
104		13	20	12	DAG	25–27, 38, 39, 41, 42, 44, 45, 61, 66, 69
105		6	120	18	DAG	0, 20, 25, 28, 37, 40, 43, 59, 60, 64, 65, 67–69, 72, 76, 78, 79
106		9	60	15	DAG	0, 25, 28, 37, 40, 43, 59, 60, 64, 65, 67, 68, 76, 78, 79
107		2	120	11	*	20, 25, 46, 51, 64, 67, 69, 70, 72, 78, 79
108		4	120	8	DAG	20, 25, 64, 67, 69, 72, 78, 79

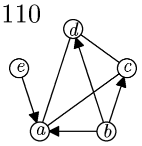
109



1 60 6 *

25, 64, 67, 70, 78, 79

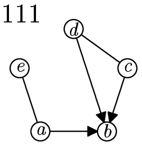
110



6 60 5 DAG

25, 64, 67, 78, 79

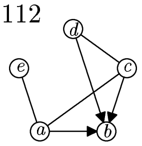
111



9 60 24 DAG

1, 6, 10, 12, 26, 27, 29, 30, 37, 40, 42, 43, 45, 49, 51, 59, 60, 64, 65, 67, 68, 76, 78, 79

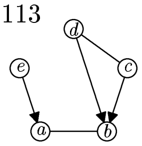
112



20 120 15 DAG

12, 37, 40, 43, 45, 51, 59, 60, 64, 65, 67, 68, 76, 78, 79

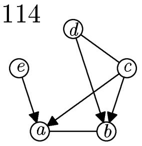
113



3 60 9 *

26, 27, 42, 45, 48, 50, 71, 72, 76

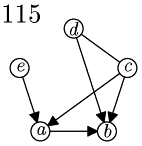
114



3 120 6 *

26, 27, 42, 45, 72, 76

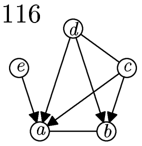
115



3 120 10 DAG

12, 26, 27, 42, 45, 51, 59, 68, 76, 79

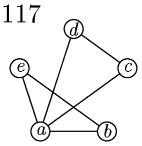
116



6 60 5 DAG

26, 27, 42, 45, 76

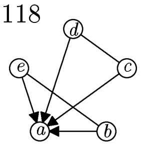
117



120 15 16 DAG

5, 14, 21, 22, 40, 43, 55, 57, 64, 65, 67, 68, 74, 75, 78, 79

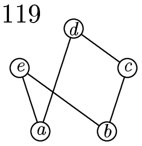
118



9 15 16 DAG

2, 7, 13, 15, 26, 27, 31, 32, 41, 42, 44, 45, 56, 58, 66, 69

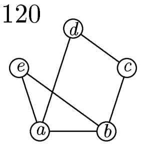
119



21 12 15

19, 22, 46, 47, 49, 58, 64, 66–68, 70, 71, 75, 78, 79

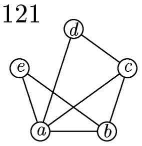
120



48 60 10

19, 22, 64, 66–68, 71, 75, 78, 79

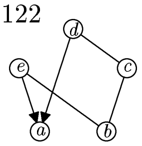
121



108 60 7 DAG

22, 64, 67, 68, 75, 78, 79

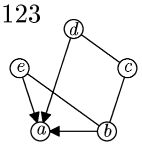
122



20 60 15 DAG

15, 19, 41, 44–47, 49, 58, 66, 69–71, 75, 78

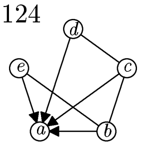
123



20 120 10 DAG

15, 19, 41, 44, 45, 58, 66, 69, 71, 78

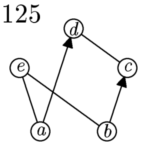
124



20 60 7 DAG

15, 41, 44, 45, 58, 66, 69

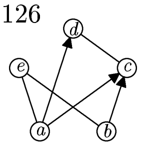
125



8 60 11 *

19, 22, 28, 64, 66–68, 71, 75, 78, 79

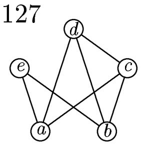
126



16 120 8 DAG

22, 28, 64, 67, 68, 75, 78, 79

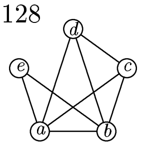
127



53 30 5

64, 67, 70, 78, 79

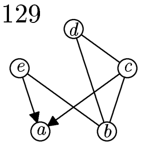
128



192 30 4 DAG

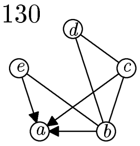
64, 67, 78, 79

129



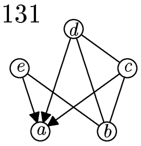
32	120	10	DAG
20, 41, 44, 46, 51, 66, 69, 70, 72, 79			

130



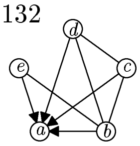
32	120	7	DAG
20, 41, 44, 66, 69, 72, 79			

131



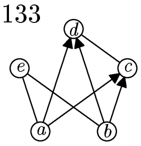
32	60	5	DAG
41, 44, 66, 69, 70			

132



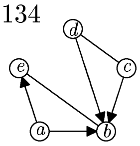
32	60	4	DAG
41, 44, 66, 69			

133



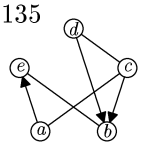
24	30	5	DAG
28, 64, 67, 78, 79			

134



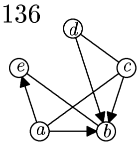
6	60	8	DAG
1, 6, 10, 12, 64, 67, 78, 79			

135



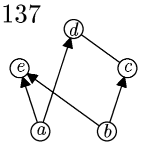
8	120	6	*
12, 64, 67, 70, 78, 79			

136



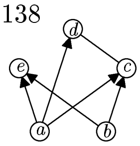
16	120	5	DAG
12, 64, 67, 78, 79			

137



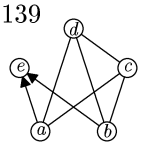
1	60	11	*
0, 19, 22, 64, 66–68, 71, 75, 78, 79			

138



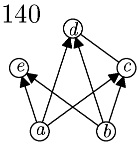
2	120	8	DAG
0, 22, 64, 67, 68, 75, 78, 79			

139



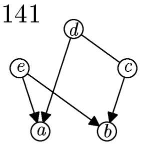
40	30	5	DAG
18, 64, 67, 78, 79			

140



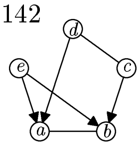
3	30	5	DAG
0, 64, 67, 78, 79			

141



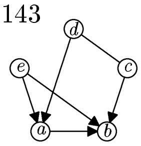
3	60	15	DAG
10, 15, 26, 27, 42, 45–47, 49, 58, 65, 69–71, 75			

142



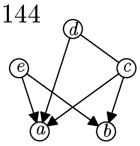
3	60	6	*
26, 27, 42, 45, 71, 75			

143



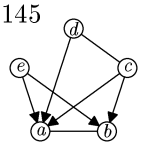
3	120	8	DAG
10, 26, 27, 42, 45, 49, 65, 75			

144



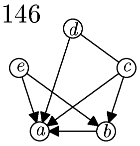
3	120	10	DAG
15, 26, 27, 42, 45, 46, 58, 69, 70, 75			

145



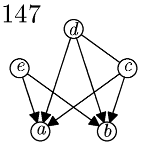
6	120	5	DAG
26, 27, 42, 45, 75			

146



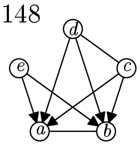
3	120	7	DAG
15, 26, 27, 42, 45, 58, 69			

147



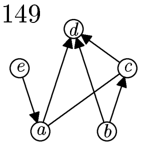
3	30	5	DAG
26, 27, 42, 45, 70			

148



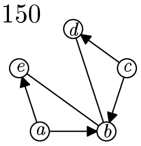
9	30	4	DAG
26, 27, 42, 45			

149



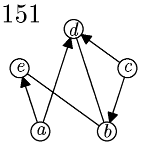
1	120	6	*
25, 46, 64, 67, 78, 79			

150



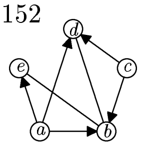
4	60	8	DAG
1, 20, 64, 67, 69, 72, 78, 79			

151



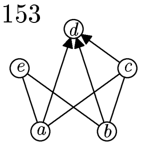
2	120	6	*
1, 64, 67, 70, 78, 79			

152



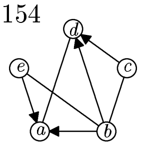
6	120	5	DAG
1, 64, 67, 78, 79			

153



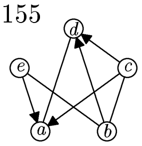
17	60	5	*
46, 64, 67, 78, 79			

154



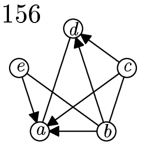
8	60	3	*
41, 71, 79			

155



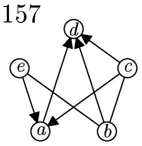
8	120	3	*
41, 70, 79			

156



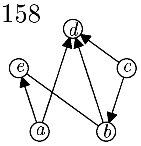
16	120	2	DAG
41, 79			

157



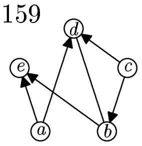
8	120	3	DAG
41, 46, 79			

158



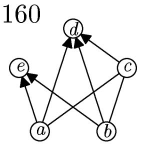
1	120	6	*
1, 46, 64, 67, 78, 79			

159



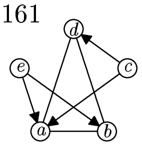
2	120	6	DAG
1, 18, 64, 67, 78, 79			

160



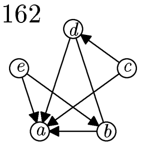
8	60	5	DAG
3, 64, 67, 78, 79			

161



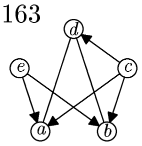
2	60	3	*
26, 74, 79			

162



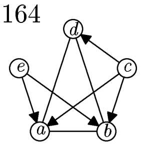
1	60	3	*
26, 56, 69			

163



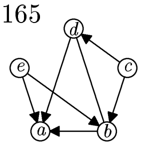
3	60	3	*
26, 70, 79			

164



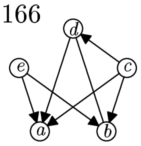
6	60	2	DAG
26, 79			

165



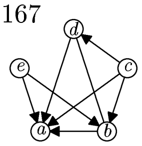
2	120	3	DAG
26, 69, 71			

166



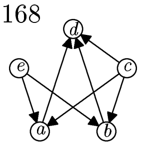
2	120	3	DAG
26, 69, 70			

167

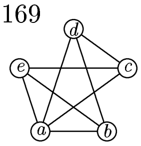
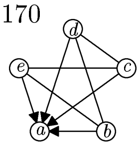
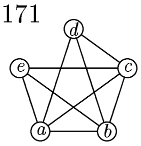
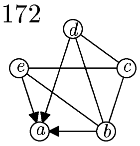
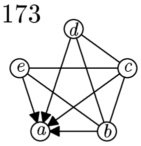
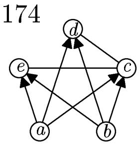
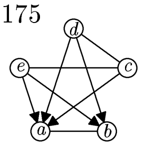
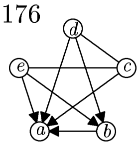
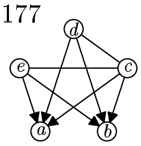
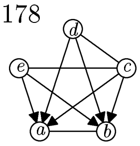
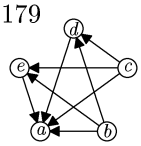
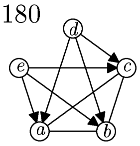
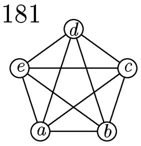


2	120	2	DAG
26, 69			

168



1	60	3	DAG
26, 46, 79			

169		82	15	2		74, 79
170		17	15	2	*	56, 69
171		248	10	1	DAG	79
172		40	60	2	DAG	69, 71
173		40	30	1	DAG	69
174		8	30	2	DAG	0, 79
175		16	60	2	DAG	45, 74
176		8	60	2	DAG	45, 56
177		8	30	2	DAG	45, 70
178		24	30	1	DAG	45
179		1	30	2	DAG	2, 69
180		13	10	1	DAG	27
181		541	1	0	DAG	

5 Conclusions

In this paper we gave a graphical characterization of the largest chain graphs of classes of Markov equivalent chain graphs which is quite clear and straightforward. The arrows in the largest chain graph can be recognized as special 'protected' arrows in every graph from the equivalence class (Consequence 3.6). Everything what one needs to examine are some special paths in the graph – complexes and descending paths between certain vertices. It provides us with a simple method for construction of the largest chain graph on basis of a given chain graph from the equivalence class (Theorem 3.9).

The given catalog of the largest chain graphs gives us the idea about the number of chain graph models over two, three, four and five variables. While in case of four variables one can check manually that the catalog is exhaustive, it is almost impossible in case of five variables. We don't know a general formula for the number of chain graph models over a given number of vertices. It remains an open question.

Let us note that one can recognize directly on basis of the largest chain graph whether the induced independency model can be described either by an undirected or a directed acyclic graph. Of course, it is an undirected graph model iff the largest chain graph is an undirected graph. An elegant characterization of chain graphs equivalent to directed acyclic graphs is given in [1], Proposition 4.2 (it appeared earlier in [9] without proof). It follows from that characterization that the models which can be described both by undirected and by directed acyclic graphs are just those models whose largest chain graph is a decomposable undirected graph.

We have indicated the directed acyclic graph models in our catalog by the mark DAG. We were also interested in pure chain graph models, that is models which cannot be described either by an undirected or by a directed acyclic graph. They are indicated by an asterisk in the catalog. In case of four variables one has 6 per cent of pure chain graph models (12 of 200) while in case of five variables one has already more than 22 per cent of pure chain graph models! One can expect that their proportion increases with the number of variables (= vertices). Perhaps it is a good argument in favor of chain graphs: they certainly allow one to describe much wider class of conditional independence structures in comparison with classic graphical approaches.

References

[1] S.A. Andersson, D. Madigan, and M.D. Perlman, On the Markov equivalence of chain graphs, undirected graphs and acyclic digraphs, *Scan. J. Stat.*, 24, 81-102, 1997.

[2] S.A. Andersson, D. Madigan, and M.D. Perlman, Alternative Markov properties for chain graphs, submitted to *Scan. J. Stat.*

[3] R.R. Bouckaert, and M. Studený, Chain graphs: semantics and expressiveness, in *Symbolic and Quantitative Approaches to Reasoning and Uncertainty* (Ch. Froidevaux, and J. Kohlas eds.), Lecture Notes in AI 946, Springer-Verlag, 67-76, 1995.

[4] W.L. Buntine, Chain graphs for learning, in *Uncertainty in Artificial Intelligence 11* (P. Besnard, and S. Hanks eds.), Morgan Kaufmann, 46-54, 1995.

[5] D.R. Cox, and N. Wermuth, Linear dependencies represented by chain graphs (with discussion), *Statist. Science*, 8, 204-283, 1993.

[6] D.R. Cox, and N. Wermuth, *Multivariate Dependencies - Models Analysis and Interpretation*, Chapman and Hall, 1996.

[7] A.P. Dawid, Conditional independence in statistical theory, *J. Roy. Statist. Soc. ser. B*, 41, 1-31, 1979.

[8] M. Frydenberg, The chain graph Markov property, *Scand. J. Statist.*, 17, 333-353, 1990.

[9] S. Højsgaard, and B. Thiesson, BIFROST - Block recursive models induced from relevant knowledge, observations, and statistical techniques, *Computat. Statist. Data Anal.*, 19, 155-175, 1995.

[10] S.L. Lauritzen, and N. Wermuth, Mixed interaction models, res. rep. R-84-8, Inst. Elec. Sys., University of Aalborg, Denmark, 1984.

[11] S.L. Lauritzen, and N. Wermuth, Graphical models for associations between variables, some of which are qualitative and some quantitative, *Ann. Statist.*, 17, 31-57, 1989.

[12] S.L. Lauritzen, Mixed graphical association models, *Scand. J. Statist.*, 16, 273-306, 1989.

[13] S.L. Lauritzen, *Graphical Models*, Clanderon Press, 1996.

[14] J. Pearl, *Probabilistic Reasoning in Intelligent Systems - Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[15] T.S. Richardson, Chain graphs and symmetric associations, in *Learning in Graphical Models* (M.I. Jordan ed.), Kluwer, 231-260, 1998.

[16] M. Studený, On separation criterion and recovery algorithm for chain graphs, in *Uncertainty in Artificial Intelligence 12* (E. Horvitz, and F. Jensen eds.), Morgan Kaufmann, 509-516, 1996.

[17] M. Studený, On recovery algorithm for chain graphs, *Int. J. Approx. Reasoning*, 17, 265-293, 1997.

[18] M. Studený, and R.R. Bouckaert, On chain graph models for description of conditional independence structures, *Ann. Statist.*, 26, 1434-1495, 1998.

[19] M. Volf, Conditional independence models induced by chain graphs (in Czech), Student work, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Prague, November 1996.

[20] N. Wermuth, and S.L. Lauritzen, On substantive research hypotheses, conditional independence graphs and graphical chain models, *J. Roy. Stat. Soc. B*, 52, 21-50, 1990.

[21] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*, John Wiley, 1990.