Linköping Studies in Science and Technology. Thesis No. 1647

Licentiate Thesis

# A Study of Chain Graph Interpretations

by

**Dag Sonntag**


Linköping University

Department of Computer and Information Science
Linköping University
SE-581 83 Linköping, Sweden

Linköping 2014

Linköping Studies in Science and Technology. Thesis No. 1647

Licentiate Thesis

# A Study of Chain Graph Interpretations

by

**Dag Sonntag**


Linköping University

Department of Computer and Information Science
Linköping University
SE-581 83 Linköping, Sweden

Linköping 2014

This is a Swedish Licentiate's Thesis

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree.
A Doctor's degree comprises 240 ECTS credits (4 year of full-time studies).
A Licentiate's degree comprises 120 ECTS credits.



ISBN 978-91-7519-377-9
ISSN 0280–7971
Printed by LiU Tryck 2014

URL: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-105024

# Abstract

Probabilistic graphical models are today one of the most well used architectures for modelling and reasoning about knowledge with uncertainty. The most widely used subclass of these models is Bayesian networks that has found a wide range of applications both in industry and research. Bayesian networks do however have a major limitation which is that only asymmetric relationships, namely cause and effect relationships, can be modelled between its variables. A class of probabilistic graphical models that has tried to solve this shortcoming is chain graphs. It is achieved by including two types of edges in the models, representing both symmetric and asymmetric relationships between the connected variables. This allows for a wider range of independence models to be modelled. Depending on how the second edge is interpreted this has also given rise to different chain graph interpretations.

Although chain graphs were first presented in the late eighties the field has been relatively dormant and most research has been focused on Bayesian networks. This was until recently when chain graphs got renewed interest. The research on chain graphs has thereafter extended many of the ideas from Bayesian networks and in this thesis we study what this new surge of research has been focused on and what results have been achieved. Moreover we do also discuss what areas that we think are most important to focus on in further research.

*This work is funded by the Swedish Research Council (ref. 2010-4808).*

# Acknowledgements

The academic world is a wonderful world full of interesting discussions and interesting people with interesting ideas. I would like to thank all the people sharing it with me, but of course there are some that I am more thankful to than others.

First and foremost I would like to express my gratitude towards my Ph.D. supervisor Jose M. Peña. He has been a large inspiration and role model both as researcher and as a person. It is he who have openened my eyes for probabilistic graphical models and who has guided me in my mental development these recent years. So thank you for all your feedback and discussions Jose!

Secondly I would like to thank my secondary supervisor, professor Nahid Shahmehri, for her help making me a better researcher. This has meant giving me a hard time when I need it and support when that is what I need. I would also like to thank her for shielding me from outside requests and giving me time for my research.

In addition to this I would like to thank the IDA administrative personnel, especially Karin and Anne, for their help in figuring out the bureaucracy of the university. Without their help I would not have managed to figure out the process of writing a travel report, let alone the process of publishing this thesis. Thank you for all your assistance! Then there are of course my colleagues and lunch buddies here at ADIT with whom I have had some really strange, but amazingly interesting, conversations. You have really made this workplace a wonderful division to work in!

Finally I would like to thank my friends and family for their support and encouragement. It is nice to get your perspective both on the outside world, as well as the academic world. The world is never boring when you are around.

So thank you all! I truly hope the time ahead of us will be just as good as it has been so far!

*Dag Sonntag*
*March 2014*
*Linköping, Sweden*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Throughout history, humans have used various models to describe natural and artificial systems in its surroundings. We will in this thesis look into one such class of models called probabilistic graphical models (PGMs). PGMs are based on the idea that the state of the variables in a system is uncertain (probabilistic) and that the interactions between variables in the system can be described according to a graph. Uncertainty can be due to several factors, the most important are that only parts of the system might be observable and that measurements might be noisy. Representing the model as a graph allows us to represent our knowledge about the system and the interactions between its variables in an intuitive manner. We can thereafter use this graph to reason and do inference when for example parts of the system state are observed. PGMs were introduced at the beginning of the last century with Wrights' path analysis [35] and Gibbs' applications to statistical physics [8]. The area then got renewed interest in computer science in the 1980s with the research of Pearl [23] and PGMs are today used in multiple applications in industry as well as society as a whole. The main advantages of using PGMs compared to other models are that the representation is intuitive, inference can be done efficiently and efficient learning algorithms exist. This has led PGMs to arguably become the most important architecture for reasoning with uncertainty [15, p.106]

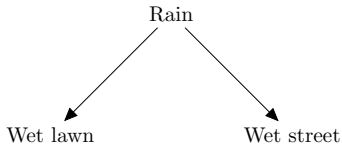
To model a system as a PGM, we first need to identify the variables of interest in it. Depending on the nature of the variables they can be modelled differently. The most researched cases are when the variables are either discrete, i.e. each variable can be in one of a finite number of states, or continuous, i.e. each variable takes a value in a continuum. The graph of a PGM does then represent these variables as nodes and the relationships between the variables as edges. Different subclasses of PGMs give different meaning to the edges. In addition to the graph a PGM class can also contain some parametrization of the variables in the model given the graph. The parameters define the probability that a variable takes a certain state or

value depending on the state or value of its neighbouring variables in the graph. The parametrization is typically represented as tables for discrete variables and as functions for continuous variables. Hence we can say that the graph of a PGM represents what variables interact in the modelled system, while the parametrization represents how they interact. An example of a PGM is shown in Fig. 1.1 which will be explained in detail later.

A PGM can be constructed in different ways. The most common are either by an expert, i.e. constructing the graph and parameters from existing knowledge, or through a learning algorithm with observational data. The observational data is then mostly in the form of samples containing the states or values of the variables for different individuals.

One of the most basic subclasses of PGMs is Markov networks (MNs). The graph of a MN is undirected and each undirected edge represents that the two variables connected by the edge are directly interacting with each other. The most well know and widely used PGM class is however Bayesian networks (BNs). A BN consist of a directed acyclic graph (DAG) in which the directed edges can be seen to represent cause and effect relationships between the variables. As an example of a BN consider the following three variables: Whether it has been raining during the night or not, whether the lawn is wet in the morning or not and whether the street is wet in the morning or not. In this case it is quite clear that the rain causes the lawn and street to become wet and hence modelling the system as a BN would result in a DAG as shown in Fig. 1.1a. We can then, given either experience or past measurements, say that the probability that it has been raining any given day is 0.3 and that the probability that the lawn is wet if it has been raining is 0.9 while it is only 0.05 if it has not been raining. Similarly we can say that the probability that the street is wet given that it has been raining is 0.8 (it dries faster than the lawn) while it is only 0.05 if it has not been raining. These conditional probability tables are shown in Fig. 1.1b.

Using this BN model we can now answer simple queries like *What is the probability that the lawn is wet given that it has been raining?* but we can also compute more advanced implicit probabilities such as the answer to *At any morning, given no other information, what is the probability that the lawn is wet?* or *If the lawn is wet, what is the probability that the street is wet?* Just looking at the DAG we can also conclude when observations about certain variables may affect the probabilities of other variables taking certain states or values. We can for example, using the DAG shown in Fig. 1.1a, see that observing the state of the wet lawn may change our belief of the state of the wet street variable if we have not observed whether it has been raining or not. The explanation for this is that by observing that the lawn is wet our belief that it has been raining may change, which in turn may change our belief that the street is wet. Hence we say that the wet street variable may be dependent on the wet lawn variable given no other information. If we on the hand have observed that it has been raining then observing that the lawn is wet does not affect our belief of the state of the




(a) The DAG $G$

| Rain | True | False |
|---|---|---|
| | 0.3 | 0.7 |

| Wet Lawn | True | False |
|---|---|---|
| Rain = True | 0.9 | 0.1 |
| Rain = False | 0.05 | 0.95 |

| Wet Street | True | False |
|---|---|---|
| Rain = True | 0.8 | 0.2 |
| Rain = False | 0.05 | 0.95 |

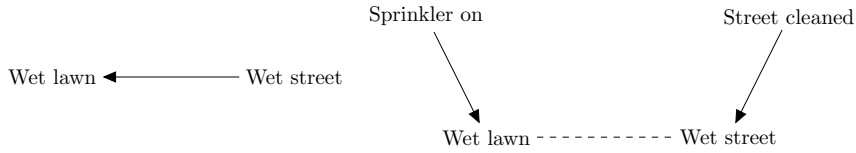
(b) Conditional probability tables

Figure 1.1: A simple Bayesian network

wet street variable. This is because observing that the street is wet does not change our belief of whether it has been raining, since we already know this. Hence we say that the wet street variable is independent of the wet lawn variable given the rain variable. How this can be read from the graph is covered in the next chapter. The important thing here is that we can, from just studying the graph, conclude which variables may be dependent and which are independent on which other variables given a third set of variables. This is why the set of all conditional independences that can be read from the graph is called the independence model represented by that graph.

As noted above BNs work fine and are widely used in different applications today ranging from error diagnostics in printers to modelling protein structures in bioinformatics or decision support systems in market analysis. BNs do however have some shortcomings due to the fact that they only model asymmetric causal relationships between variables. This means that when we want to model a system with some other kind of relationship between its variables, such as a symmetric relationship, the representation falls short. That such other kind of relationships may exist in a system can be seen in various ways like for example:

- It may be impossible for an expert of the domain, who understands the dynamics of the system, to denote one variable as the cause of the other or vice versa even though the variables are correlated.
- Intervening in the system may not support that one variable is the cause of the other variable even though they are correlated.
- The independence model of the variables in the system, i.e. all the conditional dependences and independences that exist in the system, may not be perfectly represented as a BN.




(a) Representable independence model

(b) Extended unrepresentable independence model

Figure 1.2: Non-causal relationships

To exemplify this we can take the system described for Fig. 1.1 but where we only are aware of, and have measurements for, the wet lawn and wet street variables but not the rain variable. Hence the rain variable does not exist in our model. We then know that the wet lawn and wet street variable are correlated, i.e. when we observe that the lawn is wet this increases our belief that the street also is wet and vice versa. At the same time we actually know the dynamics of the system and thereby that it is wrong to say that the wet lawn variable is the cause of the wet street or vice versa. We can also see this by intervening in the system. If we for example make the street wet by throwing water on it, this does not increase the probability that the lawn becomes wet. Nor does making the lawn wet cause the street to be wet. If we finally look at the independence model of the described system we can in this simple example note that it does not contain any conditional independences. This means that the independence model can be perfectly represented with a BN, i.e. with a BN representing all and only the conditional independences in the independence model. Such a BN is shown in Fig. 1.2a. However, if we expand the model to include two additional variables, a sprinkler on variable that indicates if the sprinkler has been on causing the lawn to be wet and a street cleaned variable indicating that the street recently has been cleaned causing the street to be wet, then the system can no longer be perfectly represented as a BN [28]. A model including the relations described in the extended system is shown in Fig 1.2b where the unrepresentable relation is shown as a dashed line.

Today systems containing non-causal relationships are however primarily modelled as BNs. This poses some problems. Firstly, the BN model becomes hard to understand and accept for an expert of the domain since it does not correspond with the known dynamics of the system. This also means that the conclusions drawn from the model about the dynamics of the underlying system might be wrong. Secondly, intervening in the system might give unexpected consequences compared to the model. In a gene regulatory example we might for example try to affect one gene to make a second gene take a certain state if we have modelled this as that the first gene is the cause of the second gene. However, if the first gene is not really the cause of the second gene, then we will not see the same effect in reality

as we do in our model, similarly as throwing water on the street does not cause the lawn to be wet.

The problems discussed above can in some cases be accepted under certain conditions. If we for example want a model of a system to only do computations on (not study in terms of dynamics) and we only make observations (no interventions), then a BN model could be used to model the system. However, from a technical point of view, it might still be a bad idea to use a BN model if the independence model cannot be perfectly represented as a BN. This is because for a BN to be able to correctly represent a system it needs to contain only the conditional dependencies that exist in the independence model of that system. Hence, if the independence model cannot be perfectly expressed as a BN any BN modelling it will need to contain some more conditional dependences, and hence fewer conditional independences, than what exist in the underlying system. By containing fewer conditional independences than those that exist in the underlying system, the advantages of using a PGM model are weakened. Hence, the model will need more data to be learnt correctly, become harder to understand and slower to do calculations on.

To solve this problem different approaches have been used. In the example above we have a hidden common cause (the rain variable) between the wet lawn and wet street variables. Hence, we can try to model this hidden common cause with a hidden node. This is performed by adding an extra node to the model that represents the unmeasured hidden common cause. Modelling hidden variables is a research field in itself and will not be covered in this thesis. Enough to say is that adding hidden nodes to a model is no trivial task. Moreover there exist other relationships between variables that cannot be solved in this manner. So in this thesis we will instead describe another approach, namely to use a more expressive PGM class called chain graphs (CGs). CGs contain a second type of edge, in addition to the directed edge, which allows a second type of relationship between variables to be modelled and thereby a much larger set of independence models to be represented compared to BNs [29]. This allows CGs to correctly model a wider range of systems [16] in a compact way that is, at the same time, interpretable, efficient to perform inference on and for which efficient learning algorithms exist. CGs were introduced in the late eighties but lately got renewed interest when more advanced systems, such as gene networks, began being modelled.

Depending on the interpretation of the second type of edge, a CG can represent different relations between variables and thereby independence models. Today there exist several possible interpretations of CGs with different separation criteria, i.e. different ways of reading conditional independences from the graph. The first interpretation (LWF) was introduced by Lauritzen, Wermuth and Frydenberg [7, 11] to combine BNs and MNs. The second interpretation (AMP) was introduced by Andersson, Madigan and Perlman to also combine BNs and MNs but with a separation criterion

more close to the one of BNs [1]. A third interpretation, the multivariate regression (MVR) was introduced by Cox and Wermuth [4] combining BNs and covariance graphs (covGs). While other interpretations have been proposed (see, for example, Drton [6]), the three interpretations above have received the most attention in the literature. They have different properties, but they are all characterised by having chain components in which the nodes are connected to each other by undirected edges (for LWF and AMP CGs) or bidirected edges (for MVR CGs). The chain components are then themselves connected to each other by directed edges.

In this thesis we give a survey of the research field of CGs. In the next chapter we give the background of the field and introduce the terminology. We will thereafter, in Chapter 3 discuss the current research in the field and how far it has come. This is followed by a short conclusion and an outline of important future research areas in Chapter 4. Finally, in Chapter 5, we describe our contribution to the field.

# Chapter 2

# Background

In the last chapter we gave a motivation to why PGMs and CGs are useful partly from a philosophical standpoint in terms of causality and intervention. In this chapter we will take a more technical standpoint and discuss PGMs and CGs in terms of representable independence models. The chapter also gives a short introduction to the research areas discussed later in the thesis. For a more complete introduction to PGMs the reader is referred to the work by Koller and Friedman [9].

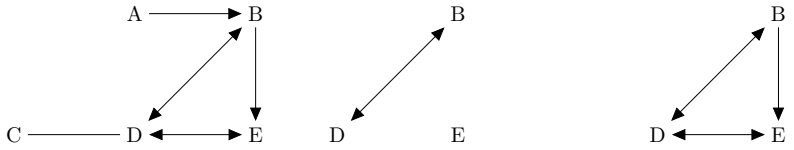
The rest of the chapter is organized as follows. First we will cover the basic notation used for PGMs and define the terms we use throughout the thesis. This is followed by a section where we discuss the advantages and disadvantages of CGs and how CGs relate to other PGM classes. The remainder of the chapter is then devoted to explaining PGMs as factorizations of probability distributions.

## 2.1 Basic notation

In this section, we review some common concepts for probabilistic graphical models (PGMs) used throughout this thesis. All graphs and probability distributions are defined over a finite set of variables $V$ represented as nodes in the graph. With $|V|$ we mean the number of variables in the set $V$ and with $V_G$ we mean the set of variables in a graph $G$.

If a graph $G$ contains an edge between two nodes $V_1$ and $V_2$, we denote with $V_1 \to V_2$ a *directed edge*, with $V_1 \leftrightarrow V_2$ a *bidirected edge* and with $V_1 - V_2$ an *undirected edge*. By $V_1 \circ\!\!\to V_2$ we mean that either $V_1 \to V_2$ or $V_1 \leftrightarrow V_2$ is in $G$. By $V_1 \multimap V_2$ we mean that either $V_1 \to V_2$ or $V_1 - V_2$ is in $G$. By $V_1 \circ\!\!-\!\!\circ V_2$ we mean that there is an edge between $V_1$ and $V_2$ in $G$.

The *parents* of a set of nodes $X$ of $G$ is the set $pa_G(X) = \{V_1 | V_1 \to V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *children* of $X$ is the set $ch_G(X) = \{V_1 | V_2 \to V_1$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *spouses* of $X$ is the set $sp_G(X) = \{V_1 | V_1 \leftrightarrow V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *neighbours* of $X$ is




(a) A graph $G$ (b) A subgraph of $G$ over $\{B, D, E\}$ (c) A subgraph of $G$ induced by $\{B, D, E\}$

Figure 2.1: A graph with 5 varibles

the set $nb_G(X) = \{V_1 | V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *boundary* of $X$ is the set $bd_G(X) = pa_G(X) \cup nb_G(X) \cup sp_G(X)$. The *adjacents* of $X$ is the set $ad_G(X) = \{V_1 | V_1 \to V_2, V_1 \leftarrow V_2$, $V_1 \leftrightarrow V_2$ or $V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$.

A *route* from a node $V_1$ to a node $V_n$ in $G$ is a sequence of nodes $V_1, \ldots, V_n$ such that $V_i \in ad_G(V_{i+1})$ for all $1 \leq i < n$. A *path* is a route containing only distinct nodes. The length of a path is the number of edges in the path. A path is called a *cycle* if $V_n = V_1$. A path is *descending* if $V_i \in pa_G(V_{i+1}) \cup sp_G(V_{i+1}) \cup nb_G(V_{i+1})$ for all $1 \leq i < n$. The *descendants* of a set of nodes $X$ of $G$ is the set $de_G(X) = \{V_n |$ there is a descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. A path is *strictly descending* if $V_i \in pa_G(V_{i+1})$ for all $1 \leq i < n$. The *strict descendants* of a set of nodes $X$ of $G$ is the set $sde_G(X) = \{V_n |$ there is a strictly descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. The *ancestors* (resp. *strict ancestors*) of $X$ is the set $an_G(X) = \{V_1 | V_n \in de_G(V_1), V_1 \notin X, V_n \in X\}$ (resp. $san_G(X) = \{V_1 | V_n \in sde_G(V_1), V_1 \notin X, V_n \in X\}$). Note that the definition for strict descendants given here coincides to the definition of descendants given by Richardson [24]. A cycle is called a *semi-directed cycle* if it is descending and $V_i \to V_{i+1}$ is in $G$ for some $1 \leq i < n$. A *subgraph* of $G$ is a subset of nodes and edges in $G$. A subgraph of $G$ *induced* by a set of its nodes $X$ is the graph over $X$ that has all and only the edges in $G$ whose both ends are in $X$.

To exemplify these concepts we can study the graph $G$ with 5 nodes shown in Fig. 2.1. In the graph we can see that $B$ is a child of $A$, $D$ is a spouse of both $B$ and $E$ while it is the neighbour of $C$. $E$ is a strict descendant of $A$ due to the strictly descending path $A \to B \to E$, while $D$ is not. $D$ is however in the descendants of $A$ together with $B, C$ and $E$. $A$ is therefore an ancestor of all variables except itself. We can also see that $G$ contains a semi-directed cycle $B \to E \leftrightarrow D \leftrightarrow B$. In Fig. 2.1b we can see a subgraph of $G$ with the variables $B, D$ and $E$ while we in Fig. 2.1c see the subgraph of $G$ induced by the same variables.

All graphs considered in this thesis are *loopless graphs*, i.e. no node can have an edge to itself. An *undirected graph* (UG) contains only undirected edges while a *covariance graph* (covG) contains only bidirected edges. A *directed acyclic graph* contains only directed edges and no semi-directed cycles. A *chain graph* (CG) under the Lauritzen-Wermuth-Frydenberg (LWF) in-

terpretation, denoted LWF CG, contains only directed and undirected edges but no semi-directed cycles. Likewise a CG under the Andersson-Madigan-Perlman (AMP) interpretation, denoted AMP CG, is a graph containing only directed and undirected edges but no semi-directed cycles. A CG under the multivariate regression (MVR) interpretation, denoted MVR CG, is a graph containing only directed and bidirected edges but no semi-directed cycles. A *chain component* $C$ of a LWF CG or an AMP CG (resp. MVR CG) is a maximal set of nodes such that there exists a path between every pair of nodes in $C$ containing only undirected edges (resp. bidirected edges). A *marginal AMP CG* (MAMP CG) is a graph containing undirected, directed and bidirected edges but with some restrictions on what structures these can take. Note that a MAMP CG is not a CG in the traditional sense since it contains three types of edges. An *ancestral graph* (AG) contains bidirected, undirected and directed edges but no subgraphs of the form $X \multimap Y - Z$ nor any pair of nodes $X$ and $Y$ st $Y \in sde(X)$ and $X \in sp_G(Y) \cup ch_G(Y)$. A *regression CG* is an AG containing no semi-directed cycles.

Let $X$, $Y$, $Z$ and $W$ denote four disjoint subsets of $V$. We say that $X$ is *conditionally independent* from $Y$ given $Z$ if the value of $X$ does not influence the value of $Y$ when the values of the variables in $Z$ are known, i.e. $p(X, Y|Z) = p(X|Z)p(Y|Z)$ holds. We denote this by $X \perp_p Y|Z$ if it holds in a probability distribution $p$. Given two independence models $M$ and $N$, we denote by $M \subseteq N$ that if $X \perp_M Y|Z$ then $X \perp_N Y|Z$ for every $X$, $Y$ and $Z$. We say that $M$ is a graphoid if it satisfies the following properties: Symmetry $X \perp_M Y|Z \Rightarrow Y \perp_M X|Z$, decomposition $X \perp_M Y \cup W|Z \Rightarrow X \perp_M Y|Z$, weak union $X \perp_M Y \cup W|Z \Rightarrow X \perp_M Y|Z \cup W$, contraction $X \perp_M Y|Z \cup W \wedge X \perp_M W|Z \Rightarrow X \perp_M Y \cup W|Z$, and intersection $X \perp_M Y|Z \cup W \wedge X \perp_M W|Z \cup Y \Rightarrow X \perp_M Y \cup W|Z$. An independence model $M$ is also said to fulfill the composition property iff $X \perp_M Y|Z \wedge X \perp_M W|Z \Rightarrow X \perp_M Y \cup W|Z$.

In a graph $G$ we say that $X$ is separated from $Y$ given $Z$ if the *separation criterion* of $G$ represents that $X$ is conditionally independent of $Y$ given $Z$ and denote this by $X \perp_G Y|Z$. The separation criteria for the different PGM classes discussed in this thesis are the following: If $G$ is a BN, covG, MVR CG, AG or regression CG then $X$ and $Y$ are separated given $Z$ iff there exists no Z-open path between $X$ and $Y$. A path is said to be *Z-open* in a BN, covG, MVR CG, AG or regression CG iff every non-collider on the path is not in $Z$ and every collider on the path is in $Z$ or $san_G(Z)$. A node $B$ is said to be a *collider* in a BN, covG, MVR CG, AG or regression CG $G$ between two nodes $A$ and $C$ on a path if the following configuration exists in $G$: $A \multimap B \leftarrow\!\circ\, C$. For any other configuration the node $B$ is a non-collider. Moreover the collider is said to be unshielded if $A$ and $C$ are non-adjacent. If $G$ is a LWF CG then $X$ and $Y$ are separated given $Z$ iff there exists no Z-open route between $X$ and $Y$. A route is said to be Z-open in a LWF CG iff every node in a non-collider section on the route is not in $Z$ and some node in every collider section on the route is in $Z$ or $an_G Z$. A *section* of a route is a maximal (wrt set inclusion) non-empty set of nodes $B_1 ... B_n$

such that the route contains the subpath $B_1 - B_2 - \ldots - B_n$. It is called a *collider section* if $B_1 \ldots B_n$ together with the two neighbouring nodes in the route, $A$ and $C$, form the subpath $A \rightarrow B_1 - B_2 - \ldots - B_n \leftarrow C$. For any other configuration the section is a non-collider section. If $G$ is an AMP CG or MAMP CG then $X$ and $Y$ is separated given $Z$ iff there exists no Z-open path between $X$ and $Y$. A path is said to be Z-open in an AMP CG or MAMP CG$G$ iff every non-head-no-tail node on the path is not in $Z$ and every head-no-tail node on the path is in $Z$ or $san_G(Z)$. A node $B$ is said to be a *head-no-tail* in an AMP or MAMP CG $G$ between two nodes $A$ and $C$ on a path if one of the following configurations exist in $G$: $A \circ\!\!\rightarrow B \leftarrow\!\!\circ C$, $A \circ\!\!\rightarrow B - C$ or $A - B \leftarrow\!\!\circ C$.

A probability distribution $p$ is said to fulfill the *global Markov property* with respect to a graph $G$, if for any $X \perp_G Y|Z$, given the separation criterion for the PGM-class to which $G$ belongs, $X \perp_p Y|Z$ holds. The *independence model* $M$ induced by a probability distribution $p$ (resp. a graph $G$), denoted as $I(p)$ (resp. $I(G)$), is the set of statements $X \perp_p Y|Z$ (resp. $X \perp_G Y|Z$) that hold in $p$ (resp. $G$). We say that a probability distribution $p$ is *faithful* to a graph $G$ when $X \perp_p Y|Z$ iff $X \perp_G Y|Z$ for all $X$, $Y$ and $Z$. We say that two graphs $G$ and $H$ are *Markov equivalent* or that they are in the same *Markov equivalence class* iff $I(G) = I(H)$. A graph $G$ is *inclusion optimal* for a probability distribution $p$ if $I(G) \subseteq I(p)$ and if there exists no other graph $H$ in the PGM class of $G$ such that $I(G) \subset I(H) \subseteq I(p)$.

## 2.2 PGM classes

PGM classes differ in what edges they contain, the separation criterion used and what structures their graphs can contain. Hence they differ in what independence models, and thereby systems, they can represent. Depending on what independence models a PGM class can represent we can discuss its expressivity. We say that a PGM class is more expressive than another class if it can express more independence models. The more basic PGM classes, such as BNs and MNs, can represent relatively few independence models for any number of nodes and and hence are not so expressive. The more general PGM classes, such as AGs, can on the other hand represent relatively many independence models and hence are very expressive.

Using an expressive PGM class has both advantages and disadvantages. The main advantage is that a model of a more expressive class is more likely to capture the true relations between the variables in the system while less expressive classes makes assumptions like for example that only causal relations exist between variables. The disadvantage of using an expressive class is that it can be harder to find the correct model since the number of possible models is much larger. This also makes it easier to overfit the learning data. Hence, to get an accurate model, more data is generally needed when learning expressive PGM classes compare to less expressive classes. Graphs with multiple types of edges can also be harder to interpret

since the interpretation of what an edge represents is not always clear. In addition to this the more basic classes, such as BNs and MNs, have received more attention in research and hence more efficient learning and inference algorithms exist for these compared to the more general classes.

A CG containing only directed edges is actually a BN, which means that any independence model that can be represented by a BN can be represented by a CG. Similarly any independence model represented by a MN (resp. covG) can be represented by a LWF or AMP CG (resp. MVR CG). This means that BNs is a subclass of all CG interpretations while MNs resp. covGs are subclasses of LWF and AMP CGs resp. MVR CGs as shown in Fig 2.2.[1] All CGs are loopless graphs but apart from this they do not share any well studied superclasses. MVR CGs are however a subclass of regression chain graphs, introduced by Wermuth and Sadeghi [34], that are part of the subtree of AGs and ribbonless graphs. Some research has also been performed on joining different CG interpretations and this has given rise to the PGM class MAMP CGs. This class of graphs contains directed, bidirected and undirected edges and is a superclass of AMP CGs and MVR CGs.

One important question when discussing different PGM classes is why CGs are interesting when there exist more general and expressive PGM classes such as loopless graphs or AGs? This has to do with the advantages and disadvantages of using more general PGM classes as discussed above. We want to be able to represent a larger set of independence models without having to suffer the disadvantages. The first disadvantage, that it can be harder to find the correct model with a larger set of possible models cannot be avoided. It simply comes with having a larger set of representable independence models. The other disadvantages can however be mitigated with further research. Many of the ideas for BNs in terms of algorithms etc. can be extended to other PGM classes and this extension is more straightforward for PGM classes similar to BNs such as CGs. It is also easier to reason about the interpretation of edges when only two types of edges exist and the graph contains no semi-directed cycles.

## 2.3 PGMs as factorizations of probability distributions

A PGM induces a factorization of a joint probability distribution of the state of a system according to its graph. If we look at the example shown in Fig. 1.1 we can see that the joint probability distribution it represents can be factorized as $p(Rain, WetStreet, WetLawn) = p(WetStreet|Rain)p(WetLawn|Rain)p(Rain)$ using the independences represented in the graph. Factorizing a large joint probability distribution has many benefits. It illuminates the conditional independences between the variables in the distribution. This means, as noted

[1] For PGM classes not defined in this thesis please check the work by Sadeghi [26].

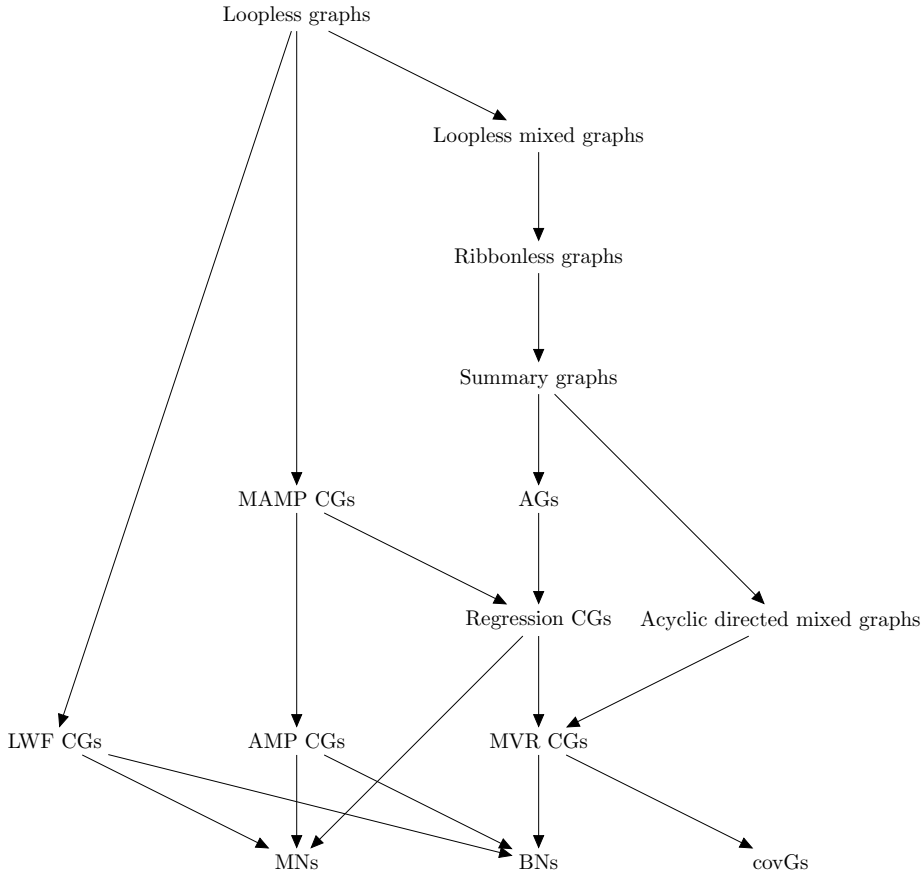



Figure 2.2: The hierarchy of PGM models

| | **Rain = True** | | **Rain = False** | |
|---|---|---|---|---|
| | **Wet Lawn = True** | **Wet Lawn = False** | **Wet Lawn = True** | **Wet Lawn = False** |
| **Wet Street = True** | 0.21600 | 0.02400 | 0.00175 | 0.03325 |
| **Wet Street = False** | 0.05400 | 0.00600 | 0.03325 | 0.63175 |

Table 2.1: A joint probability distribution

in the introduction, that the state or value of each variable only is dependent on the states or values of the neighbouring variables in the PGM graph. By interpreting the different edges in the PGM we can also deduce what kind of relations the variables have to each other. If we for example have the edge $Rain \rightarrow WetStreet$ in a BN we can interpret this as if the rain variable may be the cause of the wet street variable. Hence the graph allows us to deduce a possible explanation of the dynamics in the underlying system in a way that is not possible in a non-factorized probability distribution. To illustrate this we can compare the joint probability distribution in Table 2.1 and the DAG shown in Fig. 1.1a. The DAG does in this case correspond to a valid factorization of the joint probability distribution and hence a possible explanation of the dynamics of its underlying system. These dynamics can be seen by interpreting the DAG in a way that is not possible by looking at the joint probability distribution. Hence, factorizing a probability distribution allows us to draw conclusions about it and its underlying system. Factorizing a large joint probability distribution also means that we get multiple smaller probability distributions. This allows for efficient use of space since the size of a joint probability distribution grows exponentially with the number of nodes while the total size of local probability distributions only grows quasi-linear if most variables are conditionally independent. Multiple small probability distributions also allows us to do calculations fast.

The factorization of a probability distribution might however be performed in multiple ways, each corresponding to a different graph. These graphs do thereby represent different dynamics of the underlying system, and different understandings of how the system works. If we continue our example, we can in Fig. 2.3 see three different DAGs corresponding to different factorizations of the probability distribution shown in Table 2.1. We can here note that not all DAGs represent the conditional independence $WetLawn \perp WetStreet|Rain$, like for example the DAG in Fig. 2.3c. Generally we are however interested in the graphs representing as many of, but only, the conditional independences present in the independence model of the probability distribution, i.e. the inclusion optimal graphs. This is because modelling as many conditionally independences as possible optimizes the benefits of using PGMs described above. Note however that there might exist multiple graphs representing such independence models, as shown by the DAGs in Fig. 2.3a and 2.3b in our example.

Finding an inclusion optimal graph for a probability distribution is called

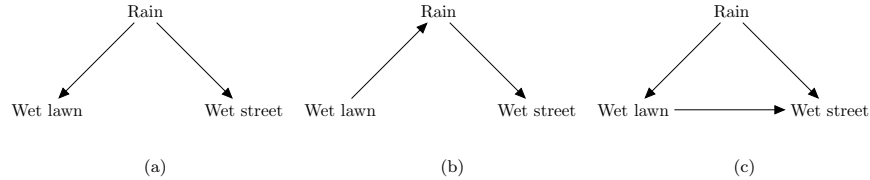



Figure 2.3: Possible DAGs representing factorizations of the probability distribution in Table 2.1

structure learning and is a well studied problem for PGMs. The input is usually a set of independent samples of the state of a system and the goal is to find the graph structure that encodes as many of, but only, the conditional independences that exist in the data. Once the structure, i.e. factorization, is learnt, the parameters can be learnt using a parameter learning algorithm. Then the model can be used to reason about the underlying system. By interpreting the edges in the PGM graph the dynamics of the system can be understood and by performing inference the probability of different variable states or values can be estimated when other variables are observed in the system.

# Chapter 3

# Current state of research

The research of CGs started in the late eighties early nineties with Lauritzen, Wermuth and Frydenberg who combined BNs and MNs to create a more expressive PGM class. The field did however fall dormant and instead the research in the PGM field was focused towards BNs. Lately though, CGs have received renewed attention and major advancements have been made. Why this renewed interest can only be speculated but important factors might be that more advanced systems are modelled and that the model creation of these has become more data driven than expert driven. This means that uncertain, and non-causal, relations might exist between the variables in the systems since the dynamics in the systems are unknown. This is in contrast to the early used BNs where the dynamics of the underlying systems were more or less known and the models were created by experts in the field.

In this chapter we discuss the recent advancements in the research field of CGs. The chapter is divided into four sections; intuition and representation, representable independence models, unique representations and finally structure learning algorithms. Each section presents the advancements made for CGs within that part of the field. One part of the PGM field that the reader might be missing is parametrization and parameter learning. We have chosen not to include this part since, although some parametrizations exist for LWF and MVR CGs, there still do not, to the authors' knowledge, exist any closed loop equations for learning these parameters. For this subfield we instead refer the reader to the work by Peña et al. [17, 18] for the LWF CG interpretation and to the work by Bergsma and Rudas [3] for the MVR CG interpretation.

## 3.1 Intuition and representation

One important question when discussing different PGM classes as representatives of independence models is *Does the independence models exist in*

*reality?* In other words, do there exist systems whose variables build up the independence models that can be represented by the PGM class? Each CG interpretation was initially motivated from a data generation perspective where each chain component could be sampled given its parents. The variables in the same chain component were then said to be on *equal footing* and it meant that these variables had symmetric relationships between them [1, 4]. For continuous variables with normally distributed errors this sampling process follows Equation 3.1 where $X$ are the nodes in the chain component that is being sampled given its parents $pa_G(X)$ in the CG $G$. $\epsilon$ represents the noise and the difference between the CG interpretations is how this noise and the $\beta$-vector are modelled. This also gives rise to the different separation criteria and different intuitive meanings for the edges in the different CG interpretations.

$$X = \beta\, pa_G(X) + \epsilon \tag{3.1}$$

### 3.1.1 LWF CGs

If we start with the LWF CG interpretation some of the first research into how the CG edges could be interpreted was done by Lauritzen and Richardson in 2002 [10]. They showed that the undirected edge in a LWF CG corresponds to a feedback relationship between two variables when they are sampled in their equilibrium state. Hence, the intuitive meaning of the undirected edge is that the nodes in the same chain component arrive at a stochastic equilibrium, being determined by their parents, as time goes to infinity. It is however unclear if this is the only interpretation and intuitive meaning behind the undirected edge in a LWF CG.

Another way to see LWF CGs is as an intersection of independence models represented by a set of BNs under conditioning [21]. This means that we have a set of different causal models that are subject to selection bias and if this bias is modelled in a certain way the intersection of all the models together form a LWF CG.

### 3.1.2 AMP CGs

Unlike in the LWF CGs the undirected edges in the AMP CGs have not been found to represent any intuitive relationship such as the feedback relationship. Any AMP CG can however be seen as to correspond to a causal model subjected to marginalization and conditioning [20]. Marginalizing away a variable means that the variable is removed from the model and that the state or value of the the variable is unknown. Conditioning out a variable also means that the variable is removed from the model, but in this case we know the state or value of the variable in the original model. Note also that the theory for transforming any AMP CG into its corresponding BN only is valid if we include certain deterministic variables in the BN, which is a rather strong assumption [20].

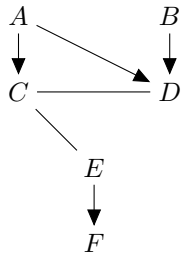



Figure 3.1: An example CG $G$

By looking at the separation criteria we can make some interesting observations. We can here see that, given no other information, any node in a chain component only is dependent on its parents, not the parents of the whole component like in LWF CGs. This means that the children of a parent of a component work as an interface between the parent and the other nodes in the component. If we for example look at the CG in Fig. 3.1 and interpret this as an AMP CG we see that $E$ is conditionally independent of $A$ and $B$ when $C$ and $D$ are unobserved.

Finally it has also been shown that just like LWF CGs the AMP CGs can be seen as an intersection of independence models represented by a set of BNs under conditioning [21]. The difference compared to LWF CGs is how the different BNs are connected and what undirected edges that are added between the different models.

### 3.1.3 MVR CGs

Unlike the other CG interpretations the bidirected edge in a MVR CG has a strong intuitive meaning. It can be seen to represent one or more hidden common causes between the variables connected by it as we saw in the example in the introduction [5]. In other words, in a MVR CG any bidirected edge $X \leftrightarrow Y$ can be replaced by $X \leftarrow H \rightarrow Y$ to obtain a BN representing the same independence model over the original variables, i.e. excluding the new variables $H$. These variables are called hidden, or latent, and have been marginalized away in the CG model [20].

## 3.2 Representable independence models

Since any CG containing only directed edges can be seen as a BN it is clear that any independence model represented by a BN can be represented by a CG. The opposite does however not hold, so a natural question is *How expressive are the CG interpretations?* It has been shown that all CG interpretations can represent some independence models only representable by that interpretation. Hence the space of all independence models repre-

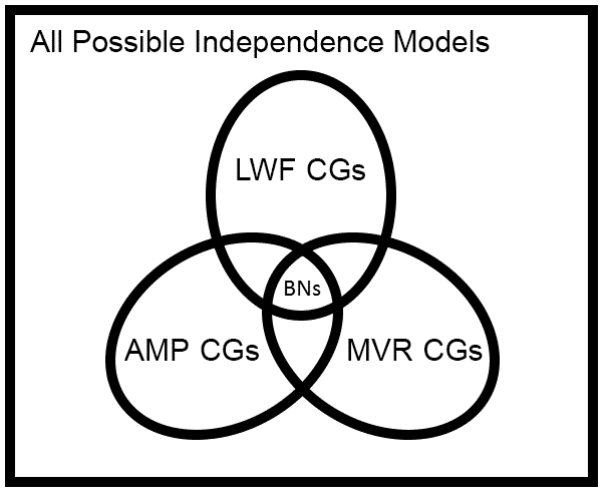



Figure 3.2: Representable independence models

sentable by CGs takes the form shown in Fig. 3.2. It has also been shown when a CG of one interpretation can be represented by a CG of another interpretation[2] and that the independence models representable by all three interpretations are those representable by BNs [28].

So how much more expressive are CGs compared to BNs? If the ratio between the number of independence models, and thereby systems, representable by BNs and those by CGs is large then the benefit of using CGs compared to BNs would maybe not be worth the difficulties. If the ratio on the other hand is small then the gain would be significant and worth the trouble. To calculate this ratio we only need to check whether each independence model representable by a CG is representable by a BN for each number of variables. This can be done by enumerating each independence model representable by CGs and such studies have been done for LWF CGs and MVR CGs for up to 5 nodes [29, 33]. The results are shown in Tables 3.1 and 3.2. For a larger number of nodes enumeration of representable independence models is no longer feasible in reasonably time. The ratio can however be approximated using a Markov chain Monte Carlo (MCMC) sampling method of the representable independence models. This method allows for approximation of the ratio for a much larger number of nodes and the results are also shown in Table 3.1 for LWF CGs and Table 3.2 for MVR CGs [29]. Using this approach it was shown that the ratio of independence models representable by LWF or MVR CGs that can be represented by BNs falls exponentially with the number of nodes and that the ratio is less than 1/1000 for more than $\approx 20$ nodes as seen in Tables 3.1 and 3.2. Hence a

[2]With the exception of when a LWF CG can be represented as an AMP CG.

significantly larger number of systems can be modelled perfectly if CGs are used compared to if BNs are used. It can also be noted from the tables that the ratios of independence models representable by CGs that also are representable by MNs resp. covGs are almost non-existent for more than 6 nodes. Finally the study showed that MVR CGs can represent a larger number of independence models compared to LWF CGs [29].

For AMP CGs no similar study has yet been performed to the authors' knowledge. There are however no indications that the results would differ significantly from those of LWF CGs or MVR CGs.

## 3.3 Unique representations

Like many other PGM classes such as BNs and AGs there might exist multiple CGs, of the same interpretation, that belong to the same Markov equivalence class as was discussed in Section 2.3. In many occasions we are however interested in the representable independence models and not the CGs themselves. This can for example be in a study such as in the previous section, but also when we are constructing learning algorithms since there exist fewer representable independence models than CGs.

Hence we are interested in having a unique graph for each representable Markov equivalence class. We would also like to have some characteristics for these graphs so that a graph can be checked whether it is such a unique representative or not. Furthermore we would also like to have a transformation algorithm to get the unique representative for a CG.

Today such representatives exist for all three interpretations. For the LWF CGs they are called *largest chain graph* (LCG) and is the CG in each Markov equivalence class that contains the maximum number of undirected edges [7]. LCGs been characterized and an algorithm for transforming any LWF CG to a LCG has been given [12, 33]. In addition to this, every LCG is a LWF CG and hence can be reasoned about as such.

For the AMP CGs there exist two different unique representatives today. These are the *maximally deflagged CGs* [25] and the *essential AMP CGs* [2]. Both of these take the form of AMP CGs and hence can be reasoned about as such. The former is based on the idea that the graph should contain as few *flags*, i.e. induced subgraphs of the form $X \to Y - Z$, as possible and secondly contain as few directed edges as possible. The essential AMP CGs are on the other hand based on the idea that any edge $X \to Y$ is in the essential AMP CG only if $X \leftarrow Y$ does not exist in any AMP CG in the Markov equivalence class that the essential AMP CG represents. Both representatives have been characterized and there exist transformation algorithms for transforming any AMP CG into either representative [2, 25].

For the MVR CG interpretation the unique representatives are called essential MVR CGs [29]. Unlike for the LWF and AMP interpretations the essential MVR CG are not actually MVR CGs. Instead, it contains the same adjacencies as any MVR CG in the Markov equivalence class with an

Table 3.1: Exact and approximate ratios of independence models representable by LWF CGs representable by MNs, BNs, neither (in that order)

| **NODES** | **EXACT** | | | **APPROXIMATE** | | |
|---|---|---|---|---|---|---|
| 2 | 1.00000 | 1.00000 | 0.00000 | 1.00000 | 1.00000 | 0.00000 |
| 3 | 0.72727 | 1.00000 | 0.00000 | 0.71883 | 1.00000 | 0.00000 |
| 4 | 0.32000 | 0.92500 | 0.06000 | 0.31217 | 0.93266 | 0.05671 |
| 5 | 0.08890 | 0.76239 | 0.22007 | 0.08093 | 0.76462 | 0.21956 |
| 6 | | | | 0.01650 | 0.58293 | 0.40972 |
| 7 | | | | 0.00321 | 0.41793 | 0.57975 |
| 8 | | | | 0.00028 | 0.28602 | 0.71375 |
| 9 | | | | 0.00018 | 0.19236 | 0.80746 |
| 10 | | | | 0.00001 | 0.12862 | 0.87137 |
| 11 | | | | 0.00000 | 0.08309 | 0.91691 |
| 12 | | | | 0.00000 | 0.05544 | 0.94456 |
| 13 | | | | 0.00000 | 0.03488 | 0.96512 |
| 14 | | | | 0.00000 | 0.02371 | 0.97629 |
| 15 | | | | 0.00000 | 0.01518 | 0.98482 |
| 16 | | | | 0.00000 | 0.00963 | 0.99037 |
| 17 | | | | 0.00000 | 0.00615 | 0.99385 |
| 18 | | | | 0.00000 | 0.00382 | 0.99618 |
| 19 | | | | 0.00000 | 0.00267 | 0.99733 |
| 20 | | | | 0.00000 | 0.00166 | 0.99834 |
| 21 | | | | 0.00000 | 0.00105 | 0.99895 |
| 22 | | | | 0.00000 | 0.00079 | 0.99921 |
| 23 | | | | 0.00000 | 0.00035 | 0.99965 |
| 24 | | | | 0.00000 | 0.00031 | 0.99969 |
| 25 | | | | 0.00000 | 0.00021 | 0.99979 |

Table 3.2: Exact and approximate ratios of independence models representable by MVR CGs representable by covGs, BNs, neither (in that order)

| NODES | EXACT | | | APPROXIMATE | | |
|---|---|---|---|---|---|---|
| 2 | 1.00000 | 1.00000 | 0.00000 | 1.00000 | 1.00000 | 0.00000 |
| 3 | 0.54545 | 1.00000 | 0.00000 | 0.72547 | 1.00000 | 0.00000 |
| 4 | 0.10714 | 0.82589 | 0.10714 | 0.28550 | 0.82345 | 0.10855 |
| 5 | 0.00807 | 0.59074 | 0.36762 | 0.06967 | 0.59000 | 0.36787 |
| 6 | | | | 0.01241 | 0.40985 | 0.57921 |
| 7 | | | | 0.00187 | 0.28675 | 0.71145 |
| 8 | | | | 0.00028 | 0.19507 | 0.80465 |
| 9 | | | | 0.00002 | 0.13068 | 0.86930 |
| 10 | | | | 0.00000 | 0.08663 | 0.91337 |
| 11 | | | | 0.00000 | 0.05653 | 0.94347 |
| 12 | | | | 0.00000 | 0.03771 | 0.96229 |
| 13 | | | | 0.00000 | 0.02385 | 0.97615 |
| 14 | | | | 0.00000 | 0.01592 | 0.98408 |
| 15 | | | | 0.00000 | 0.00983 | 0.99017 |
| 16 | | | | 0.00000 | 0.00644 | 0.99356 |
| 17 | | | | 0.00000 | 0.00485 | 0.99515 |
| 18 | | | | 0.00000 | 0.00267 | 0.99733 |
| 19 | | | | 0.00000 | 0.00191 | 0.99809 |
| 20 | | | | 0.00000 | 0.00112 | 0.99888 |
| 21 | | | | 0.00000 | 0.00073 | 0.99927 |
| 22 | | | | 0.00000 | 0.00048 | 0.99952 |
| 23 | | | | 0.00000 | 0.00035 | 0.99965 |
| 24 | | | | 0.00000 | 0.00017 | 0.99983 |
| 25 | | | | 0.00000 | 0.00014 | 0.99986 |

arrowhead on an edge if and only if every MVR CG in the Markov equivalence class contains an arrowhead on that edge. This definition is similar to that of essential graphs for BNs and AGs but it also means that there might exist undirected edges in the essential MVR CGs. However, using the same separation criterion as for MVR CGs shown in Section 2.1, an essential MVR CG represents the same independence model as the MVR CGs' it is representative for [29]. Essential MVR CGs have been characterized and there does also exist a transformation algorithm that allows any MVR CG to be transformed into its essential MVR CG [29].

In addition to a unique representation of the representable independence models we might also be interested in exploring what CGs there exist in a certain Markov equivalence class. In other words we would like to, given a CG, see what other CGs that exist in that Markov equivalence class. This is possible using the so called split and merging operators that can transform any CG into another CG of the same Markov equivalence class through a sequence of steps. Today such operators exist for all CG interpretations [27, 28, 32]. The names comes from their way to split or merge different chain components with each other by replacing undirected or bidirected edges with directed edges or vice versa. The operators then describe the conditions for when a split or a merging of two adjacent chain components is possible without altering the Markov equivalence class of the graph.

## 3.4 Structure learning algorithms

As discussed in the previous chapter finding algorithms that learn an inclusion optimal graph from data is important. Today there exist mainly two approaches to the problem, the constraint based approach and the score based approach. The constraint based approach checks for conditional independences in the data using different independence tests such as the $\chi^2$ test. The score based approach on the other hand uses a score function measuring the likelihood of the data given the structure. Today there exist efficient learning algorithms for both approaches for the more basic PGM classes, such as BNs and MNs, while we in the more general classes, such as CGs, are restricted to the constraint based approach. This is due to the difficulty of finding fast and efficient score functions.

One common constraint based approach to the structural learning problem is that of the PC algorithm for BNs [14, 30]. The algorithm is based on three sequential steps. In the first step the adjacencies of the graph are found. In the second step these adjacencies are oriented into directed edges according to a set of rules. These rules are applied repeatedly until no rule is applicable and results in a so called *essential graph* which, if interpreted as a LWF CG, represents the correct independence model. The third step then orients the remaining undirected edges so that the graph becomes a BN. Today there exist PC like algorithms for all three CG interpretations [19, 27, 31] where the rules and the last step are replaced according to the

interpretation. An example of the PC like algorithm for MVR CGs can be seen in Algorithm 1 with its corresponding rules in Fig. 3.3 [27]. The algorithm learns, given a probability distribution $p$ faithful to an unknown MVR CG $G$, a MVR CG $H$ such that $I(H) = I(G)$. We can here see that line 1 to 7 finds the adjacencies in the graph (step 1). Line 8 and 9 orient these according to a set of rules (step 2) while the remaining lines orient the remaining edges into directed edges without creating any new unshielded colliders (step 3). The PC like algorithms for CGs are proven to learn a CG with the correct independence model if the probability distribution of the data is faithful to some CG of the chosen CG interpretation. However, if this is not the case, it can be shown that the learnt model might not be inclusion optimal with respect to the independence model of the data [22].

1 Let $H$ denote the complete undirected graph
2 For $l = 0$ to $l = |V_H| - 2$
3  Repeat while possible
4  Select any ordered pair of nodes $A$ and $B$ in $H$ st $A \in ad_H(B)$ and $|ad(A) \setminus B| \geq l$
5  If there exists a $S \subseteq (ad_H(A) \setminus B)$ st $|S| = l$ and $A \perp_p B|S$ then
6  Set $S_{AB} = S_{BA} = S$
7  Remove the edge $A - B$ from $H$
8 Apply rule 0 while possible
9 Apply rules 1-3 while possible
10 Let $H_u$ be the subgraph of $H$ containing only the nodes and the undirected edges in $H$
11 Let $T$ be the clique tree of $H_u$
12 Order the cliques $C_1, ..., C_n$ of $H_u$ st $C_1$ is the root of $T$ and if $C_i$ is closer to the root than $C_j$ in $T$ then $C_i < C_j$.
13 Order the nodes st if $A \in C_i$, $B \in C_j$ and $C_i < C_j$ then $A < B$
14 Orient the undirected edges in $H$ according to the ordering obtained in line 13
15 Return $H$

**Algorithm 1:** PC like learning algorithm for MVR CGs

For the AMP and MVR CG interpretations the PC like algorithms are, to the authors' knowledge, the only learning algorithms defined so far. For the LWF interpretation two other learning algorithms do however exist, both constraint based. The first is called the LCD algorithm and is based on a divide and conquer approach [13]. This algorithm requires, as the PC like algorithms, the probability distribution of the data to be faithful for it to learn an inclusion optimal CG. The second algorithm, called CKES, do however relax this prerequisite [22]. The CKES algorithm starts with the empty graph and then iteratively improves the graph to fit the data better. In each iteration the algorithm checks if some edge can be removed from the graph without decreasing the fit or if some edge can be added to improve the fit. The algorithm also replaces the current graph with a Markov

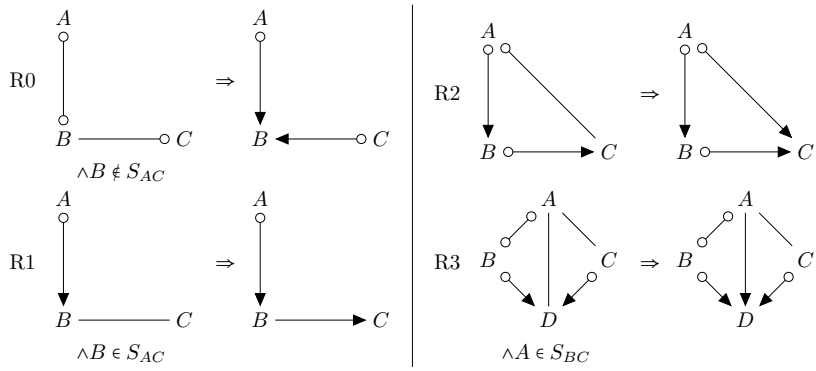



Figure 3.3: The rules for Algorithm 1

equivalent one from time to time to avoid local optima. As noted earlier the CKES algorithm does not require the probability distribution of the data to be faithful to learn the inclusion optimal CG. Instead it is enough if the distribution fulfills the graphoid properties and the composition property. It should be noted that both these conditions are required for any algorithm to learn an inclusion optimal graph efficiently [22].

# Chapter 4

# Conclusions and future work

In this thesis we have tried to give an introduction to CGs and the research presented in the area so far. We have also tried to motivate CGs through the advantages and disadvantages of using PGM classes with different expressiveness. Using a more expressive PGM class, compared to a less expressive PGM class, grants the advantage that a more accurate model might be found when modelling a system. However, the disadvantage is that there exist more possible models, and hence the best model might be harder to find. Furthermore the model might be harder to interpret and the learning algorithms might be slower since the research has come furthest in the basic, least expressive, PGM classes such as BNs or MNs.

We have in this thesis shown that the advantage of using CGs compared to today's commonly used BNs is considerably in the sense that only a small fraction of the systems representable by CGs can be represented by BNs accurately. We have also shown that with the advancements made in recent years in the research field, like for example efficient structure learning algorithms, the disadvantages of using CGs has shrunk. The research has by no means come as far as for BNs, but many of the necessary elements do today exist for using CGs in practice. This has, in the authors meaning, made CGs to be a viable choice of PGM class when modelling advanced systems with uncertain relations between its variables.

Some work does however remain before CGs can get widespread use outside academia. Most important here might be that the parametrization needs to be clarified and closed loop equations for estimating the parameters need to be developed. Other important areas for research are score based structure learning algorithms and efficient learning algorithms that do not require faithfulness from the probability distribution of the data. Moreover, CGs have not yet, to the authors' knowledge, been applied with structure learning algorithms, such as those described in Section 3.4, to any large

real world problem. Having such an example could, in addition to show the advantages of CGs compared to BNs, allow for additional insight into what relations the secondary edges of the CG interpretations represent. Hence, we believe that partly shifting focus from theory to more practical examples could be greatly beneficial for the research of the field.

# Bibliography


[1] S. A. Andersson, D. Madigan, and M. D. Perlman. An Alternative Markov property for Chain Graphs. *Scandianavian Journal of Statistics*, 28:33–85, 2001.

[2] S. A. Andersson and M. D. Perlman. Characterizing Markov Equivalence Classes For AMP Chain Graph Models. *The Annals of Statistics*, 34:939–972, 2006.

[3] W. P. Bergsma and T. Rudas. Marginal Models for Categorical Data. *The Annals of Statistics*, 30:140–159, 2002.

[4] D. R. Cox and N. Wermuth. Linear Dependencies Represented by Chain Graphs. *Statistical Science*, 8:204–283, 1993.

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

[6] M. Drton. Discrete Chain Graph Models. *Bernoulli*, 15:736–753, 2009.

[7] M. Frydenberg. The Chain Graph Markov Property. *Scandinavian Journal of Statistics*, 17:333–353, 1990.

[8] J. Gibbs. *Elementary Principles of Statistical Mechanics.* Yale University Press, 1902.

[9] D. Koller and N. Friedman. *Probabilistic Graphcal Models.* MIT Press, 1999.

[10] S. L. Lauritzen and T. S. Richardson. Chain Graph Models and their Causal Interpretations. *Journal of the Royal Statistical Society: Series B*, 64:321–361, 2002.

[11] S. L. Lauritzen and N. Wermuth. Graphical Models for Association Between Variables, Some of Which are Qualitative and Some Quantitative. *The Annals of Statistics*, 17:31–57, 1989.

[12] B. Liu, Z. Zheng, and H. Zhao. An Efficient Algorithm for Finding the Largest Chain Graph According to a Given Chain Graph. *Science in China Series A: Mathematics*, 48:1517–1530, 2005.

[13] Z. Ma, X. Xie, and Z. Geng. Structural Learning of Chain Graphs via Decomposition. *Journal of Machine Learning Research*, 9:2847–2880, 2008.

[14] C. Meek. Strong Completeness and Faithfulness in Bayesian networks. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 411–418, 1995.

[15] R. E. Neopolitan and X Jiang. *Contemporary Artificial Intelligence*. CRC Press, 2013.

[16] J. M. Peña. Approximate Counting of Graphical Models Via MCMC. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 352–359, 2007.

[17] J. M. Peña. Faithfulness in Chain Graphs: The Discrete Case. *International Journal of Approximate Reasoning*, 50:1306–1313, 2009.

[18] J. M. Peña. Faithfulness in Chain Graphs: The Gaussian Case. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 588–599, 2011.

[19] J. M. Peña. Learning AMP Chain Graphs under Faithfulness. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 251–258, 2012.

[20] J. M. Peña. Error AMP Chain Graphs. In *Proceedings of the 12th Scandinavian Conference on Artificial Intelligence*, pages 215–224, 2013.

[21] J. M. Peña. Every LWF and AMP Chain Graph Originates from a Set of Causal Models. arXiv:1312.2967 [stat.ML], 2013.

[22] J. M. Peña, D. Sonntag, and J. Nielsen. An Inclusion Optimal Algorithm for Chain Graph Structure Learning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, 2014. Accepted.

[23] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[24] T. S. Richardson. Markov Properties for Acyclic Directed Mixed Graphs. *Scandinavian Journal of Statistics*, 30:145–157, 2003.

[25] A. Roverato and M. Studený. A Graphical representation of Equivalence Classes of AMP Chain Graphs. *Journal of Machine Learning Research*, 7:1045–1078, 2006.

[26] K. Sadeghi. Markov Equivalences for Subclasses of Loopless Mixed Graphs. arXiv:1110.4539 [stat.OT], 2011.

[27] D. Sonntag and J. M. Peña. Learning Multivariate Regression Chain Graphs under Faithfulness. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 299–306, 2012.

[28] D. Sonntag and J. M. Peña. Chain Graph Interpretations and Their Relations. In *Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, pages 510–521, 2013.

[29] D. Sonntag, J. M. Peña, and Manuel Gómez-Olmedo. Approximate Counting of Graphical Models Via MCMC Revisited. *Under Review*, 2014.

[30] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer-Verlag, 1993.

[31] M. Studený. On Recovery Algorithms for Chain Graphs. *International Journal of Approximate Reasoning*, 17:265–293, 1997.

[32] M. Studený, A. Roverato, and Š. Štěpánová. Two Operations of Merging and Splitting Components in a Chain Graph. *Kybernetika*, 45:208–248, 1997.

[33] M. Volf and M. Studený. A Graphical Characterization of the Largest Chain Graphs. *International Journal of Approximate Reasoning*, 20:209–236, 1999.

[34] N. Wermuth and K. Sadeghi. Sequences of Regression and Their Independences. arXiv:1103.2523 [stat.ME], 2012.

[35] S. Wright. The Method of Path Coefficients. *The Annals of Mathematical Statistics*, 5:161–215, 1934.

# Chapter 5

# Our Contribution

This chapter presents our contribution to the research field of CGs. The chapter starts with a short summary of the work and this is then followed by the publications themselves. Note that all articles included here have not yet been published but are submitted and in the review process.

## 5.1 Summary

We have done research in all of the subareas of the research field described in Chapter 3. For the intuition and representation part this is mostly restricted to a book chapter we wrote as an introduction to how CGs can be used when reasoning about biomedical knowledge. The book chapter is presented in Article 1 in Section 5.2. The book itself is an attempt to connect the current research in data representation and the practice in the medical community. The chapter contains an introduction to the field of CGs motivated from a gene network modelling perspective. It does also contain an appendix where the separation criteria of the different CG interpretations are seen as systems of linear equations. For the appendix our contribution is restricted to MVR CGs.

When it comes to the research of representable independence models our contribution is more substantiate. It starts with Article 2, presented in Section 5.3, proving when an independence model of a graph of a certain CG interpretation can be represented as a graph of another CG interpretation. The article also presents and proves soundness and completeness for a feasible split operation for AMP CGs as described in Section 3.3. Moreover we have also, in Article 3, presented in Section 5.4, used the MCMC approach to approximate the ratio of independence models representable by MVR CGs that also are representable by BNs. To do this we first had to define the essential MVR CGs, their characteristics and a transformation algorithm for these new graphs as described in Section 3.3. Hence this work is also included in the article. Our contribution in the article comprised

everything regarding CGs, while the part regarding BNs was performed by the co-authors.

Finally, for the structure learning part we have presented and proved the correctness of two different learning algorithms. The first is the PC like algorithm for MVR CGs in Article 4, presented in Section 5.5. This article also contains the split and merging operations for MVR CGs described in Section 3.3 which were needed to prove the correctness of the algorithm. The second is the CKES algorithm that was described in Section 3.4 which is presented in Article 5 in Section 5.6. Our contribution for this article and algorithm mainly consists of first transforming the algorithm from theory to practice and then to implement it and do an experimental evaluation. The transformation was needed since parts of the theoretic algorithm were infeasible to perform in reasonable time and hence some approximations had to be taken.

## 5.2 Article 1: Chain graphs and gene networks

### Bibliography:

Title: Chain Graphs and Gene Networks
Authors: Dag Sonntag and Jose M. Peña
Publication: Chapter in the book Foundations for Biomedical Knowledge Representation (2014), Springer
Status: Under review

# Chain Graphs and Gene Networks

Dag Sonntag and Jose M. Peña

ADIT, IDA, Linköping University, Sweden

When modelling a biological system we are interested in the different entities it comprises, as well as their attributes and their internal dependencies. The nature of these entities and attributes varies depending on the system under examination, as described in the previous chapter.

In this chapter we will focus on genes as entities and their expression levels as attributes. In a biological system each of these attributes can be seen as a random variable and, from a probabilistic point of view, the whole system can be seen as a probability distribution. The probability distribution in turn can be seen as a set of conditional independencies and dependencies between its variables. We will show how these probability distributions and their independence structure can be modelled by the class of probabilistic graphical models (PGMs) known as chain graphs (CGs). CGs include Bayesian networks (BNs, discussed in the previous chapter) as a particular case, and thus share many of the ideas behind them.

As was the case with BNs, the fundamental idea behind CGs is to represent the independence structure of a probability distribution as a graph. There are several possible interpretations of CGs with different *separation criteria*, i.e. different ways of reading conditional independencies from the graph. The first interpretation (LWF) was introduced by Lauritzen, Wermuth and Frydenberg [7, 13] to combine Bayesian and Markov networks (MNs). The second interpretation (AMP) was introduced by Andersson, Madigan and Pearlman [1]. A third interpretation, the multivariate regression (MVR) was introduced by Cox and Wermuth [3]. While other interpretations have been proposed (see, for example, Drton [5]), the three interpretations above have received the most attention in the literature. They have different properties, but they are all characterised by chain components that are connected by directed edges. The nodes within each chain component are connected to each other by undirected edges (for LWF and AMP) or bidirected edges (for MVR).

Using multiple kinds of edges makes CGs more expressive than BNs. In turn, this allows CGs to correctly model a wider range of probability distributions and independence models [18] in a compact way that is, at the same time, easy to interpret, to learn and to perform inference on. On the one hand, this expressiveness makes it possible to formulate plausible models for most biological systems, thus lessening the risk of obscuring the true (in)dependencies in a gene network. On the other hand, it might be more difficult to find the best model. CGs can also be harder to interpret because edges do not have not have an as intuitive causal meaning as they do in a BNs. This makes them more appropriate to use in advanced systems,

such as gene networks, in which it the causal relationship between different entities can be hard to know or even non-existent. We will in this chapter present the different CG interpretations in terms of separation criteria, in terms of systems of linear equations, and finally in terms of intuitive meaning behind the edges in a CG.

The chapter is organised as follows. Section 1 describes the notation used throughout the chapter. In Section 2 we present the different interpretations of CGs, while in Section 3 we describe how a CG can be learnt from a probability distribution. After a short conclusion and summary in Section 4, we provide an alternative illustration of CGs as systems of linear models in the appendix. For simplicity, we limit our discussion to continuous variables; however, most results can be generalised to systems with discrete or mixed variables.

# 1 Background and notation

In this section, we review some concepts from probabilistic graphical models that are used later in this chapter. All graphs and probability distributions are defined over a finite set of variables $V$.

If a graph $G$ contains an edge between two nodes $V_1$ and $V_2$, we denote with $V_1 \to V_2$ a *directed edge*, with $V_1 \leftrightarrow V_2$ a *bidirected edge* (sometimes also called a *dashed edge*) , and with $V_1 - V_2$ an *undirected edge*. A set of nodes is said to be *complete* if there exists edges between all pairs of nodes in the set. A complete set of nodes is said to be a *clique* if there exists no superset of it that is complete.

The *parents* of a set of nodes $X$ of $G$ is the set $pa_G(X) = \{V_1 | V_1 \to V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *children* of $X$ is the set $ch_G(X) = \{V_1 | V_2 \to V_1$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *spouses* of $X$ is the set $sp_G(X) = \{V_1 | V_1 \leftrightarrow V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *neighbours* of $X$ is the set $nb_G(X) = \{V_1 | V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *boundary* of $X$ is the set $bd_G(X) = pa_G(X) \cup nb_G(X) \cup sp_G(X)$. The *adjacents* of $X$ is the set $ad_G(X) = \{V_1 | V_1 \to V_2$,$V_1 \leftarrow V_2$, $V_1 \leftrightarrow V_2$ or $V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$.

A *route* from a node $V_1$ to a node $V_n$ in $G$ is a sequence of nodes $V_1, \ldots, V_n$ such that $V_i \in ad_G(V_{i+1})$ for all $1 \leq i < n$. A *path* is a route containing only distinct nodes. The length of a path is the number of edges in the path. A path is called a *cycle* if $V_n = V_1$. A path is *descending* if $V_i \in pa_G(V_{i+1}) \cup sp_G(V_{i+1}) \cup nb_G(V_{i+1})$ for all $1 \leq i < n$. The *descendants* of a set of nodes $X$ of $G$ is the set $de_G(X) = \{V_n |$ there is a descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. A path is *strictly descending* if $V_i \in pa_G(V_{i+1})$ for all $1 \leq i < n$. The *strict descendants* of a set of nodes $X$ of $G$ is the set $sde_G(X) = \{V_n |$ there is a strictly descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. The *ancestors* (resp. *strict ancestors*) of $X$ is the set $an_G(X) = \{V_1 | V_n \in de_G(V_1), V_1 \notin X, V_n \in X\}$ (resp. $san_G(X) = \{V_1 | V_n \in sde_G(V_1), V_1 \notin X, V_n \in X\}$). Note that the

definition for strict descendants given here coincides to the definition of descendants given by Richardson [22]. A cycle is called a *semi-directed cycle* if it is descending and $V_i \to V_{i+1}$ is in $G$ for some $1 \leq i < n$.

A MN contains only undirected edges while a BN only contains directed edges and no semi-directed cycles. A CG under the Lauritzen-Wermuth-Frydenberg (LWF) interpretation, denoted LWF-CG, contains only directed and undirected edges but no semi-directed cycles. Likewise a CG under the Andersson-Madigan-Perlman (AMP) interpretation, denoted AMP-CG, is a graph containing only directed and undirected edges but no semi-directed cycles. A CG under the multivariate regression (MVR) interpretation, denoted MVR-CG, is a graph containing only directed and bidirected edges but no semi-directed cycles. A *chain component* $C$ of a LWF-CG or an AMP-CG (resp. MVR-CG) is a maximal set of nodes such that there exists a path between every pair of nodes in $C$ containing only undirected edges (resp. bidirected edges). A *subgraph* of $G$ is a subset of nodes and edges in $G$. A subgraph of $G$ induced by a set of its nodes $X$ is the graph over $X$ that has all and only the edges in $G$ whose both ends are in $X$.

Let $X$, $Y$ and $Z$ denote three disjoint subsets of $V$. We say that $X$ is *conditionally independent* from $Y$ given $Z$ if the value of $X$ does not influence the value of $Y$ when the values of the variables in $Z$ are known, i.e. $p(X, Y|Z) = p(X|Z)p(Y|Z)$ holds. We denote this by $X \perp_p Y|Z$ if it holds in a probability distribution $p$. Moreover we say that $X$ is separated from $Y$ given $Z$ in a graph $G$ if the separation criterion of $G$ represents that $X$ is conditionally independent of $Y$ given $Z$. We denote the this by $X \perp_G Y|Z$ and we will discuss different separation criteria for CGs later in this chapter. A probability distribution $p$ is said to fulfill the *global Markov property* with respect to a graph $G$, if for any $X \perp_G Y|Z$, given the separation criterion for the PGM-class to which $G$ belongs, $X \perp_p Y|Z$ holds. The *independence model* $M$ induced by a probability distribution $p$ (resp. a graph $G$), denoted as $I(p)$ (resp. $I(G)$), is the set of separation statements $X \perp_p Y|Z$ (resp. $X \perp_G Y|Z$). Given two independence models $M$ and $N$, we say that $N$ includes $M$ ($M \subseteq N$), iff $X \perp_M Y|Z$ implies that $X \perp_N Y|Z$ for every $X$, $Y$ and $Z$.

We say that a probability distribution $p$ is *faithful* to a graph $G$ when $X \perp_p Y|Z$ iff $X \perp_G Y|Z$ for all $X$, $Y$ and $Z$. We say that two graphs $G$ and $H$ are *Markov equivalent* or that they are in the same *Markov equivalence class* iff $I(G) = I(H)$. A graph $G$ is *inclusion optimal* for a probability distribution $p$ if $I(G) \subseteq I(p)$ and if there exists no other graph $H$ in the PGM class of $G$ such that $I(G) \subset I(H) \subseteq I(p)$.

## 2 Interpretations

The research of CGs started in the late 1980s with the Lauritzen-Wermuth-Frydenberg (LWF) interpretation in order to combine BNs and MNs into more expressive models. Subsequently, the Andersson-Madigan-Perlman

(AMP) interpretation and the multivariate regression (MVR) interpretation, both in common use in recent literature, were proposed. Each interpretation is based on a different separation criterion and a different interpretation of edges. No interpretation subsumes another [5, 25], and no interpretation is generally better than any other. LWF, AMP and MVR interpretations are just different from each other and are suited to different problems. We will in this chapter present each interpretation in the classical sense, i.e. in terms of a separation criterion as in Drton [5], but also in terms of a system of linear equations. Finally we will also try to describe the intuitive meaning for each interpretation.

In the case of BNs, the interpretation in terms of a separation criterion is as follows. Given three disjoint set of nodes $X$, $Y$ and $Z$ in a BN $G$, $X \perp_G Y | Z$ iff there exists no path between $X$ and $Y$ such that:

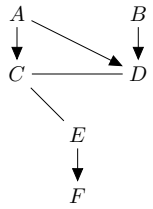
1. every non-collider on the path is not in $Z$ and
2. every collider on the path is in $Z$ or $san_G(Z)$.

A node $B$ is said to be a *collider* between two nodes $A$ and $C$ on a path if the following configuration exists in the path: $A \rightarrow B \leftarrow C$. For any other configuration the node $B$ is a non-collider on the path. In addition, the interpretation in terms of a system of linear equations is as follows. The probability distribution of every node in a BN depends only on its parents. This means that every node $X_i$ is modelled by the equation $X_i = \beta_i * pa_G(X_i) + \epsilon_i$ in the associated system of linear equations, where $\beta_i$ is a weight vector measuring the influence of the individual parents and the noise $\epsilon_i \sim \mathcal{N}(0, \sigma_i)$ is independent of any other node's noise. The intuitive meaning is simply that the parent nodes are the cause of the children nodes.

For CGs the different interpretations have different separation criteria. As noted in the introduction, the feature all CGs share is that they contain subgraphs, called chain components, that are connected to each other by directed edges. Within each chain component the type of edges varies depending on the interpretation: LWF-CGs and AMP-CGs contain undirected edges while MVR-CGs contain bidirected edges. Even though the intuitive meaning of a CG is not as simple as for a BN, there are similarities between the two PGM classes. For example, the separation statements encoded by a CG correspond to the non-existence of routes with certain features, as in BNs. Moreover, in terms of linear equations each component of a CG can be seen as a supernode, with a corresponding probability distribution determined only by its parents. If we let $K_i$ be the component $i$ in a CG $G$, then $G$ has an associated system of linear equations with normally distributed errors as follows:

$$K_i = \beta_i \, pa_G(K_i) + \epsilon_i \qquad \text{where} \qquad \epsilon_i \sim \mathcal{N}(0, \Lambda_i).$$

$\epsilon_i$ represents the noise, or influence, between the nodes in the same component. How this noise and the $\beta_i$-vector are modelled varies between the different interpretations, and gives them different intuitive meanings.




(a) An example CG $G$.

$A \perp_G B|\emptyset$
$B \not\perp_G C|\emptyset$
$A \not\perp_G B|C$
$A \perp_G F|E$
$B \not\perp_G E|D$

(b) LWF-CG separations in $G$.

$A \perp_G B|\emptyset$
$B \perp_G C|\emptyset$
$A \perp_G B|C$
$A \perp_G F|E$
$B \not\perp_G E|D$

(c) AMP-CG separations in $G$.

Figure 2.1: An example CG $G$ and some corresponding separations according to the LWF and AMP interpretations.

## 2.1 The LWF interpretation

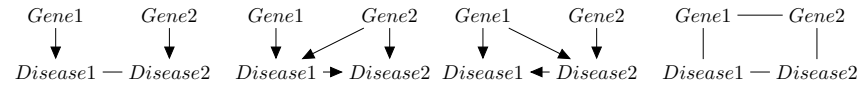
The LWF interpretation was introduced by Lauritzen, Wermuth and Frydenberg in 1989 [7, 13]. As noted above, LWF-CGs contain components that are connected to each other by directed edges. The separation criterion is the following. Given three disjoint subsets of nodes $X$, $Y$ and $Z$ in an LWF-CG $G$, $X \perp_G Y|Z$ iff there exists no route between $X$ and $Y$ such that:

1. every node in a non-collider section on the route is not in $Z$ and
2. some node in every collider section on the route is in $Z$.

A *section* of a route is a maximal non-empty set of nodes $B_1...B_n$ such that the route contains the subpath $B_1 - B_2 - \ldots - B_n$. It is called a *collider section* if $B_1 \ldots B_n$ together with the two neighbouring nodes in the route, $A$ and $C$, form the subpath $A \rightarrow B_1 - B_2 - \ldots - B_n \leftarrow C$ in the route. For any other configuration the section is a non-collider section.

A simple example of a CG is shown in Figure 2.1a. Here the CG has four chain components: $A$, $B$, $\{C, D, E\}$ and $F$. If the graph is interpreted as a LWF-CG, the separations and non-separations shown in Figure 2.1b hold. Note that these are not all the separations that hold in $G$.

When reasoning in terms of linear equations, the parents of a component can be interpreted as the causes of the nodes in that component, and directed edges have the same meaning as in a BN. On the other hand, nodes within each component are connected by undirected edges and the influences between the nodes are described by potentials, as in MNs. So the linear equation of a node $X_j$ in a LWF-CG is $X_j = \beta_j \, pa_G(K_i) + \epsilon^j$ where $K_i$ is the component to which $X_j$ belongs. This means that the $l$-th element of $\beta_j$ can be interpreted as the sum of path weights over all the paths in $G$ between parent $l$ of $K_i$ and $X_j$ through the nodes in $K_i$. The noise $\epsilon^j$ is then controlled by the corresponding value in the associated inverse covariance matrix of that component. Furthermore, the corresponding entry in the inverse covariance matrix for two nodes $X_j$ and $X_m$ can be non-zero iff




(a) Inclusion optimal CG. (b) Inclusion optimal BNs. (c) Inclusion optimal MN.

Figure 2.2: A gene and disease example with CG representation, BN representation and MN representation.

there exists an undirected edge $X_j - X_m$ in $G$ (see the Appendix for details). For example, we can see from Figure 2.1a that the influence from node $B$ onto node $D$ is direct since there only exists one open path. However, the influence from node $A$ onto node $E$ is determined by the path $A \to C - E$ as well as $A \to D - C - E$ (see the Appendix for details).

This characterisation of the influence of a parent of $K_i$ means that parents influence all the nodes in $K_i$, as influence propagates to all of $K_i$ through its undirected edges. We can see, for example, that in the second example above the influence from $A$ onto $E$ is the same as $A$ onto $C$ except for the last path between $C$ and $E$. This makes LWF-CGs similar to module networks, another PGM class that has shown promising results for gene networks [23]. In module networks every node in a module, which is similar to a component, has the same parents and parameters. In a LWF-CG, they have the same parents when the LWF-CG is seen as a system of linear equations.

A simple example from the gene modelling domain where LWF-CGs are appropriate can be a system containing two genes and two diseases caused by these such that *Gene1* is the only cause of *Disease1* and *Gene2* is the only cause of *Disease2*. Moreover let us assume that data has shown that *Disease1* and *Disease2* are correlated and that *Gene1* and *Gene2* are correlated when the state of *Disease1* and *Disease2* are known. The LWF-CG representing the information above is seen in Figure 2.2a while the corresponding inclusion optimal BNs and MN are shown in Figure 2.2b and 2.2c, respectively. As seen the information described and the independences in the system are much easier to read from the LWF-CG than from the BNs or MN.

Other settings in which LWF-CGs are appropriate, such as a system with feedback in its equilibrium state, are described by Lauritzen and Richardson [12]. Another example in the literature is given by Ferrándiz et al. [6]. They show that LWF-CGs are useful when the variables of a system only can be measured in an aggregated state. It can also be noted that if a LWF-CG only contains directed edges it can be read as a BN while if it only contains undirected edges it can be read as a MN.

## 2.2 The AMP interpretation

The AMP-CG interpretation was introduced by Andersson, Madigan and Perlman [1] as an alternative to the LWF interpretation because it preserves

the recursive characteristics of BNs. Similarly to LWF-CGs, AMP-CGs also contain components connected to each other by directed edges, whereas each component internally only contains undirected edges. As a result, an AMP-CG containing only directed edges can be read as a BN and an AMP-CG containing only undirected edges can be read as a MN.

However, the separation criterion is different compared to LWF-CGs. Given three disjoint subsets of nodes $X$, $Y$ and $Z$ in an AMP-CG $G$ $X \perp_G Y | Z$ iff there exists no path between $X$ and $Y$ such that:

1. every non-collider on the path is not in $Z$ and
2. every collider on the path is in $Z$ or $san_G(Z)$.

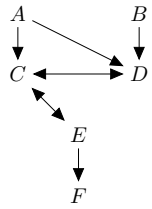
A node $B$ is said to be a *collider* in an AMP-CG $G$ between two nodes $A$ and $C$ on a path if one of the following configurations exists in $G$: $A \rightarrow B \leftarrow C$, $A \rightarrow B - C$ or $A - B \leftarrow C$. For any other configuration the node $B$ is a non-collider. In the case of the CG shown in Figure 2.1a, we can see that the separations and non-separations in Figure 2.1c hold if we interpret it as an AMP-CG. Note that these are not all the separations and non-separations that hold in $G$.

The modelling of the noise also differs from LWF-CGs. It can be shown that the associated linear equation of a node $X_j$ in an AMP-CG $G$ is $X_j = \beta_j \, pa_G(X_j) + \epsilon_j$. The node depends only on its parents of node and not on the parents of the whole component, as is the case in LWF-CGs. The noise $\epsilon_j$ is then controlled by the corresponding value in the associated inverse covariance matrix of that component. Furthermore, the corresponding entry in the inverse covariance matrix for two nodes $X_j$ and $X_k$ can be non-zero iff there exists an undirected edge $X_j - X_k$ in $G$ (see the Appendix for details). Intuitively, a small set of nodes works as an interface between other nodes in the component and its parents. For example, we can see that $C$ and $D$ in Figure 2.1a block the influence from the parents $A$ and $B$ onto $E$.

As an example from the gene modelling domain we can once again take the one described for LWF-CGs. As before the inclusion optimal AMP-CG is shown in Figure 2.2a and the inclusion optimal BNs and MNs are shown in Figure 2.2b and 2.2c. Again, the CG gives a much clearer view of the independencies in the system compared to the BNs or MN. While the LWF-CG and AMP-CG have the same structure, they represent different independence models. Which interpretation and CG that is best to use then depends on further information about the system and what model that makes most sense for the experts in the field.

## 2.3 The MVR interpretation

MVR-CGs were originally introduced by Cox and Wermuth [3, 4], and are equivalent to the acyclic directed mixed graphs without semi-directed cycles presented by Richardson [22]. Cox and Wermuth represented these graphs




(a) Example CG $G$.

$A \perp_G B | \emptyset$
$B \perp_G C | \emptyset$
$A \perp_G B | C$
$A \perp_G F | E$
$A \not\perp_G F | C$

(b) MVR-CG separations in $G$.

Figure 2.3: A MVR-CG and some corresponding separations.

using directed edges and dashed edges, but we follow Richardson [22] as we feel that the notation is closer to that of BNs when it comes to the separation criterion.

The most important difference between the MVR-CGs compared to AMP-CGs and LWF-CGs is that MVR-CG components contains bidirected instead of undirected edges. As a result, MVR-CGs have the following separation criterion: Given three disjoint subsets of nodes $X$, $Y$ and $Z$ in a MVR-CG $G$ $X \perp_G Y | Z$ iff there exists no path between $X$ and $Y$ such that:

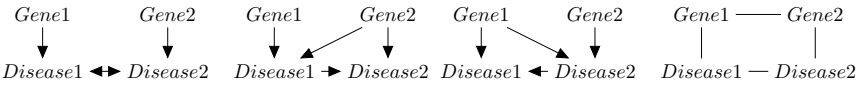
1. every non-collider on the path is not in $Z$ and
2. every collider on the path is in $Z$ or $san_G(Z)$.

A node $B$ is said to be a *collider* in a MVR-CG $G$ between two nodes $A$ and $C$ on a path if one of the following configurations exists in the path: $A \rightarrow B \leftarrow C$, $A \rightarrow B \leftrightarrow C$, $A \leftrightarrow B \leftarrow C$ or $A \leftrightarrow B \leftrightarrow C$. For any other configuration the node $B$ is a non-collider. An example of MVR-CG is shown in Figure 2.3a, with some of the corresponding separations and non-separations in Figure 2.3b. Note that a MVR-CG containing only directed edges can be read as a BN while a MVR-CG containing only bidirected edges can be read as a covariance graph [4].

The associated system of linear equations is similar to that of the AMP-CGs: each node depends only on its parents and not on the parents of the whole component. The associated linear equation for a node $X_j$ can therefore be written as $X_j = \beta_j \, Pa(X_j) + \epsilon_j$, where $\epsilon_j$ is dependent on the other nodes in the same component. Unlike AMP-CGs, MVR-CGs contain non-zero values in the corresponding covariance matrix for nodes that are spouses (see the Appendix for details). The intuitive meaning behind the MVR-CGs is therefore very close to that of AMP-CGs, differing only in the noise modelling.

A typical model that gives rise to a MVR-CG arises in the presence of hidden variables (i.e. unobserved variables that are parents of one or more observed variables) in the data. Consider once more the gene-disease example described for LWF-CGs and AMP-CGs. Let us assume that we have the same information as before, i.e. that *Gene1* (resp. *Gene2*) is the only

known cause of *Disease1* (resp. *Disease2*), that the data has shown that *Disease1* and *Disease2* are correlated and *Gene1* and *Gene2* are correlated when *Disease1* and *Disease2* are known. However, let us assume that this time we suspect the presence of an unknown factor inducing the correlation between *Disease1* and *Disease2*, such as being exposed to a stressful environment. Having such a hidden variable results in the independence model described in the information above. We can now choose whether we would like to model this hidden variable or not in our model, but due to difficulties of measurement let us assume we do not. The MVR-CG representing the information above is then shown in Figure 2.4a while the inclusion optimal BNs and MN are shown in Figure 2.4b and 2.4c, respectively. For this simple example, the AMP-CG and the MVR-CG represent the same independence model. To appreciate the difference between AMP-CGs and MVR-CGs, we could add another set of nodes (*Gene3* and *Disease3*) to the example such that *Gene3* is the only cause of *Disease3* and *Disease2* and *Disease3* are correlated.




(a) Inclusion optimal CG. (b) Inclusion optimal BNs. (c) Inclusion optimal MN.

Figure 2.4: A gene and disease example with MVR-CG representation, BN representation and MN representation.

## 2.4 Expressibility and gene networks

CGs are interesting because they can correctly represent a much larger set of independence models, and thereby probability distributions, than BNs or MNs. BNs and MNs are the two PGM classes most commonly used today when modelling bioinformatics systems. This means that for a probability distribution $p$ there may be no BN $G_{BN}$ such that $I(p) = I(G_{BN})$ but there may be a CG $G_{CG}$ such that $I(p) = I(G_{CG})$, as we have seen in the previous subsections. It should be noted that a BN can represent every probability distribution, but to do so the independence model of the graph might have to contain fewer independencies than the actual probability distribution as seen above. These spurious dependencies can then later be "removed" by the correct parametrization, but this is still problematic for several reasons. Firstly, the advantage of using PGMs, such as the speed of inference, is larger the sparser a graph is. By having more edges than necessary this advantage is lost. Secondly, some of these edges might not make sense from a biological point of view. This is problematic for practitioners trying to understand the system through its graph, since the edges obscure the true (in)dependencies between the variables.

So how much more expressive are CGs compared to BNs? If the advantage is small the additional complexity might not translate into significantly better models. So far only LWF-CGs have been investigated in this respect; Peña [18] showed that only 13% of all CGs with 10 nodes could be represented perfectly by BNs. At 13 nodes, this fraction was down to 4.1%. So for a large domain with hundreds of nodes the number of independence models representable by BNs is incredibly small compared the number of independence models representable by CGs. Therefore, CGs are much more likely to provide a realistic graph structure instead of obscuring the true relations in the system.

# 3 CG learning

As is the case for BNs, the graph structure of a CG can be learnt either from expert knowledge on the system or from data. The process of creating a CG from expert knowledge is very similar to that of a BN but where a second kind of edges can be used to model variable correlations as described in the previous section. In this section we will therefore only discuss how a CG is learnt from data and we will assume that this data is in the form of a probability distribution $p$. Only a few learning algorithms exists for CGs today: most of them are constraint based and assume that $p$ is faithful to some CG.

## 3.1 Learning CGs under the faithfulness assumption

First let us assume that the probability distribution is faithful to some CG; this assumption will be relaxed in the next subsection. We say that a probability distribution $p$ is faithful to some CG $G$ iff $G$ have the same separations and non-separations as independences and dependences in $p$, i.e. that $G$ can perfectly represent the independence model of $p$. This means that a probability distribution $p$ that is faithful to some LWF-CG $G$ not necessarily is faithful to some AMP-CG $H$, and hence that faithfulness is dependent on the PGM class we have in mind [25].

It is important to stress that this is a strong assumption. However, faithfulness allows for very fast and efficient algorithms since the reasoning in the algorithms can be made in the space of all CG models, instead of in the much larger space of all independence models. Today there exist structure learning algorithms for all three interpretations under the faithfulness assumption. Three of these are based on the PC-algorithm [15, 26] used for BNs and contain three phases. In the first phase they learn the adjacencies of the CG; in the second, some of the edges are oriented according to simple rules; and in the third, the remaining edges are oriented to avoid semi-directed cycles. This allows for an efficient way of learning the structure where no step has to be backtracked. For a comprehensive treatment of these algorithms we refer the reader to Studený's work [27] for LWF-CGs,

Peña's work for AMP-CGs [20] and Sonntag and Peña's work [24] for MVR-CGs. Finally there also exists a second, decomposition-based algorithm for learning LWF-CGs developed by Ma et al. that has been shown to be of lower complexity than the PC-variant algorithm [14].

## 3.2 Weakening the faithfulness assumption

It has been argued that it is unlikely that a randomly generated probability distribution that factorizes according to a BN is unfaithful to the BN [16]. While this is true if every parameter in a BN is generated randomly, the argument may not hold if the parameters have been hand picked (e.g. by a designer or by nature through evolution). Needless to say these are mostly the systems we are interested in modelling.

If one would apply the learning algorithms described in the previous subsection on a probability distribution that is not faithful to a CG of the appropriate interpretation it can no longer be guaranteed that the learnt CG can factorize the probability distribution properly. This means that the learnt CG might represent independences that does not exist in the underlying system which the probability distribution represents. Hence there might exist relations between variables in the underlying system that are not represented in the CG model. Moreover this means that no matter how the CG is parametrized it can never represent the original probability distribution perfectly. This is of course a problem since we would like to learn an inclusion optimal CG, i.e. a CG that can factorize the probability distribution, but contains as many separations as possible [21].

Unfortunately, learning a CG without assuming faithfulness is very complex and computationally demanding. The only algorithm for this task in the current literature is the CKES algorithm for LWF-CGs presented by Peña et al. [21], which is based on a similar algorithm for BNs called KES [17]. The algorithm works by iteratively adding resp. removing separations between variables in the CG that are independent resp. dependent in the probability distribution given their boundary in the CG of that iteration. This is performed by adding resp. removing the appropriate edges in the CG. Moreover, to ensure that an inclusion optimal CG is reached at the end of the algorithm all CGs in the Markov equivalence class of the CG in the last iteration has to be searched for improvements. Like all efficient learning algorithms certain assumptions do however have to be made about the probability distribution. These are that the independence model induced by it fulfills the graphoid properties as well as the composition property [21]. The graphoid properties are satisfied for all strictly positive probability distributions, while the composition property is satisfied for every Gaussian probability distribution.

### 3.3 Factorisation and parameter learning

Very little research has been done on CG parameter learning, and only for LWF-CGs. The factorisation of a probability distribution $p$ with variables $X_1, \ldots, X_n$ according to a LWF-CG $G$ with components $K_1, \ldots, K_m$ is

$$p(X_1, ..., X_n) = \prod_{i=1}^{m} p(K_i | pa_G(K_i)). \tag{3.1}$$

Each component $K_i$ can then be factorized clique-wisely as follows

$$p(K_i | pa_G(K_i)) = \frac{1}{Z_i} \prod_{M \in M_C} \phi_M, \tag{3.2}$$

where $M_C$ are the complete subsets in the closure graph of $K_i$, i.e. the induced subgraph $G_{K_i \cup pa_G(K_i)}$ where each arc is replaced by an undirected edge and each pair of vertices in $pa_G(K_i)$ are also connected by an undirected edge. Each $\phi_M$ is then the potential over the variables in $M$ and $Z_i$ is a normalization constant. In other words, the probability distribution of the closure graph of each component can be seen as a MN. To parametrize these products and potentials we can then simply parametrize the system of linear equations since there exist a one to one relation between it and the probability distribution.

A qualitative approach to CGs has also been introduced by Lappenschaar et al. [10]. In a qualitative CG it is only calculated whether two variables adjacent in the graph have positive, negative or ambiguous influence on each other, and not the actual parameter value. In the article Lappenschaar et al. describes the advantages of using qualitative CGs compared to qualitative BNs when modelling interaction between diseases. One of the mayor advantages were that the CGs were able to capture equilibrium models.

## 4 Conclusion

In this chapter we have shown how CGs can be used to model complex system such as gene networks. We have also tried to show some advantages of using CGs compared to using BNs or MNs, which are more commonly used in real-world applications today. CGs are more flexible because they can represent a larger class of independence models compared to BNs or MNs. This means that CGs can express a model that is closer, or at least as close, to the real system as any BN or MN. At the same time, they are still easy to interpret and one can relate their structure to the underlying molecular processes.

We have also shown that there exist structure learning algorithms for the three interpretations of CGs if the probability distribution is faithful to some CG of that interpretation. One more general structure learning algorithm, which only requires the probability distribution to fulfill the graphoid and

composition properties, has also been discussed for the LWF interpretation. Using these algorithms on an advanced system like a gene network will result in a CG which can give a good insight into how the variables interact, even when causal relations between variables can not be defined within the system.

# 5 Appendix: System of linear equations for CGs

In this appendix we derive and present how the separation criteria of the different interpretations translate into systems of linear equations.

## 5.1 LWF-CGs

Let $G$ be a LWF-CG with connectivity components $K_1, \ldots, K_n$. Let $\mathcal{N}(G)$ denote the set of regular Gaussian distributions that factorize with respect to $G$, which coincide with the set of distributions that satisfy the LWF global Markov property with respect to $G$ [11, Theorems 3.34 and 3.36]. Let $p \in \mathcal{N}(G)$. Assume without loss of generality that $p$ has mean 0. Let $\Omega^i_{K_i,K_i}$ and $\Omega^i_{K_i,pa_G(K_i)}$ denote submatrices of the precision matrix $\Omega^i$ of $p(K_i, pa_G(K_i))$. Then, as shown in [2, Section 2.3.1],

$$K_i | pa_G(K_i) \sim \mathcal{N}(\beta^i pa_G(K_i), \Lambda^i) \tag{5.1}$$

where

$$\beta^i = -(\Omega^i_{K_i,K_i})^{-1}\Omega^i_{K_i,pa_G(K_i)} \tag{5.2}$$

and

$$(\Lambda^i)^{-1} = \Omega^i_{K_i,K_i}. \tag{5.3}$$

Then, as shown in [19, Section 3], $G$ has associated a system of linear equations with normally distributed errors as follows. For every $K_i$,

$$K_i = \beta^i \, pa_G(K_i) + \epsilon^i \tag{5.4}$$

where

$$\epsilon^i \sim \mathcal{N}(0, \Lambda^i) \tag{5.5}$$

and

$$(\Omega^i_{K_i,K_i})_{j,k} = 0 \text{ for all } j,k \in K_i \text{ such that } j - k \text{ is not in } G \tag{5.6}$$

and

$$(\Omega^i_{K_i,pa_G(K_i)})_{j,k} = 0 \text{ for all } j \in K_i \text{ and } k \in pa_G(K_i) \tag{5.7}$$

such that

$$j \leftarrow k \text{ is not in } G. \tag{5.8}$$

It is worth mentioning that the mapping above between the probability distributions in $\mathcal{N}(G)$ and the systems of linear equations is bijective [19, Lemma 1]. Moreover, an alternative (but equivalent) parameterization of the probability distributions in $\mathcal{N}(G)$ is presented in [28].

Then, $G$ has associated a system of linear equations with correlated errors as follows. For every $X_j \in K_i$,

$$X_j = \beta_j \, pa_G(K_i) + \epsilon^j \tag{5.9}$$

where

$$\beta_j \text{ is the } j\text{-th row of } \beta^i \tag{5.10}$$

and

$$Cov(\epsilon^j, \epsilon^k) = (\Lambda^i)_{j,k}. \tag{5.11}$$

Note that $X_j$ is a linear combination of $pa_G(K_i)$ and not of $pa_G(X_j)$. Note also that, as shown in [29, Propositon 5.7.3],

$$(\Omega^i_{K_i,K_i})^{-1} = \Sigma_{K_i \cdot pa_G(K_i)} \tag{5.12}$$

where $\Sigma_{K_i \cdot pa_G(K_i)}$ represents the partial covariance matrix of $K_i$ given $pa_G(K_i)$.

Then, as shown in [8, Theorem 1], the element $(A, B)$ of $(\Omega^i_{K_i,K_i})^{-1}$ can be written as a sum of path weights over all the paths in $G$ between $A$ and $B$ trough nodes in $K_i$. Specifically,

$$\begin{aligned} ((\Omega^i_{K_i,K_i})^{-1})_{A,B} &= (\Sigma_{K_i \cdot pa_G(K_i)})_{A,B} \\ &= \sum_{\rho \in \rho_{A,B}} (-1)^{|\rho|+1} \frac{|(\Omega^i_{K_i,K_i})_{\setminus\rho}|}{|\Omega^i_{K_i,K_i}|} \prod_{l=1}^{|\rho|-1} (\Omega^i_{K_i,K_i})_{\rho_l,\rho_{l+1}} \end{aligned} \tag{5.13}$$

where $\rho_{A,B}$ denotes the set of paths in $G$ between $A$ and $B$ through nodes in $K_i$, $|\rho|$ denotes the number of nodes in a path $\rho$, $\rho_l$ denotes the $l$-th node in $\rho$, and $(\Omega^i_{K_i,K_i})_{\setminus\rho}$ is the matrix with the rows and columns corresponding to the nodes in $\rho$ omitted. Moreover, the determinant of a zero-dimensional matrix is taken to be 1. This leads to the following interpretation of $\beta_j$: By Equations 5.2, 5.10 and 5.13, the $k$-th element of $\beta_j$ can be written as sum of path weights over all the paths in $G$ between $X_k$ and $X_j$ trough nodes in $K_i$.

## 5.2 AMP-CGs

Let $G$ be an AMP-CG with connectivity components $K_1, \ldots, K_n$. Let $\mathcal{N}(G)$ denote the set of regular Gaussian distributions that satisfy the AMP global Markov property with respect to $G$. Let $p \in \mathcal{N}(G)$. Assume without loss of generality that $p$ has mean 0. Then, as shown above, $K_i | pa_G(K_i) \sim \mathcal{N}(\beta^i pa_G(K_i), \Lambda^i)$. Then, as shown in [1, Section 5], $G$ has associated a

system of linear equations with normally distributed errors as follows. For every $K_i$,

$$K_i = \beta^i \, pa_G(K_i) + \epsilon^i \tag{5.14}$$

where

$$\epsilon^i \sim \mathcal{N}(0, \Lambda^i) \tag{5.15}$$

and

$$((\Lambda^i)^{-1})_{j,k} = 0 \text{ for all } j, k \in K_i \text{ such that } j - k \text{ is not in } G \tag{5.16}$$

and

$$(\beta^i)_{j,k} = 0 \text{ for all } j \in K_i \text{ and } k \in pa_G(K_i) \text{ such that } j \leftarrow k \text{ is not in } G. \tag{5.17}$$

It is worth mentioning that the mapping above between the probability distributions in $\mathcal{N}(G)$ and the systems of linear equations is bijective [1, Section 5]. Moreover, the first constraint here coincides with the first constraint in the previous section.

Then, $G$ has associated a system of linear equations with correlated errors as follows. For every $X_j \in K_i$,

$$X_j = \beta_j \, pa_G(X_j) + \epsilon^j \tag{5.18}$$

where

$$\beta_j \text{ contains the nonzero elements of } (\beta^i)_j. \tag{5.19}$$

and

$$Cov(\epsilon^j, \epsilon^k) = (\Lambda^i)_{j,k}. \tag{5.20}$$

Note that, unlike in the previous section, $X_j$ is here a linear combination of $pa_G(X_j)$ and not of $pa_G(K_i)$.

## 5.3 MVR-CGs

Let $G$ be a MVR-CG with connectivity components $K_1, \ldots, K_n$. Then, $G$ has associated a system of linear equations with normally distributed errors as shown in the previous section except for two differences. First, $\mathcal{N}(G)$ now denotes the set of regular Gaussian distributions that satisfy the MVR global Markov property with respect to $G$. Second, we now replace Equation 5.16 with

$$(\Lambda^i)_{j,k} = 0 \text{ for all } j, k \in K_i \text{ such that } j \leftrightarrow k \text{ is not in } G. \tag{5.21}$$

See also [9].

# Acknowledgements

This work is funded by the Center for Industrial Information Technology (CENIIT) and a so-called career contract at Linköping University, and by the Swedish Research Council (ref. 2010-4808).

# References

[1] S. A. Andersson, D. Madigan, and M. D. Perlman. An Alternative Markov property for Chain Graphs. *Scandianavian Journal of Statistics*, 28:33–85, 2001.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] D. R. Cox and N. Wermuth. Linear Dependencies Represented by Chain Graphs. *Statistical Science*, 8:204–283, 1993.

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

[5] M. Drton. Discrete Chain Graph Models. *Bernoulli*, 15:736–753, 2009.

[6] J. Ferrándiz, E. F. Castillo, and P. Snamartín. Temporal Aggregation In Chain Graph Models. *Journal of Statistical Planning and Inference*, 133:69–93, 2005.

[7] M. Frydenberg. The Chain Graph Markov Property. *Scandinavian Journal of Statistics*, 17:333–353, 1990.

[8] B. Jones and M. West. Covariance Decomposition in Undirected Gaussian Graphical Models. *Biometrika*, 92:779–786, 2005.

[9] C. Kang and J. Tian. Markov Properties for Linear Causal Models with Correlated Errors. *Journal of Machine Learning Research*, 10:41–70, 2009.

[10] M. Lappenschaar, A. Hommersom, and P. J. F. Lucas. Qualatitive Chain Graphs and their Use in Medicine. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 179–186, 2012.

[11] S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

[12] S. L. Lauritzen and T. S. Richardson. Chain Graph Models and their Causal Interpretations. *Journal of the Royal Statistical Society: Series B*, 64:321–361, 2002.

[13] S. L. Lauritzen and N. Wermuth. Graphical Models for Association Between Variables, Some of Which are Qualitative and Some Quantitative. *The Annals of Statistics*, 17:31–57, 1989.

[14] Z. Ma, X. Xie, and Z. Geng. Structural Learning of Chain Graphs via Decomposition. *Journal of Machine Learning Research*, 9:2847–2880, 2008.

[15] C. Meek. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403–410, 1995.

[16] C. Meek. Strong Completeness and Faithfulness in Bayesian networks. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 411–418, 1995.

[17] J. D. Nielsen, T. Kočka, and J. M. Peña. On Local Optima in Learning Bayesian Networks. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 435–442, 2003.

[18] J. M. Peña. Approximate Counting of Graphical Models Via MCMC. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 352–359, 2007.

[19] J. M. Peña. Faithfulness in Chain Graphs: The Gaussian Case. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 588–599, 2011.

[20] J. M. Peña. Learning AMP Chain Graphs under Faithfulness. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 251–258, 2012.

[21] J. M. Peña, D. Sonntag, and J. Nielsen. An Inclusion Optimal Algorithm for Chain Graph Structure Learning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, 2014. Accepted.

[22] T. S. Richardson. Markov Properties for Acyclic Directed Mixed Graphs. *Scandinavian Journal of Statistics*, 30:145–157, 2003.

[23] E. Segal, D. Pe'er, A. Regev, D. Koller, and N. Friedman. Learning Module Networks. *Journal of Machine Learning Research*, 6:557–588, 2005.

[24] D. Sonntag and J. M. Peña. Learning Multivariate Regression Chain Graphs under Faithfulness. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 299–306, 2012.

[25] D. Sonntag and J. M. Peña. Chain Graph Interpretations and Their Relations. In *Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, pages 510–521, 2013.

[26] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer-Verlag, 1993.

[27] M. Studený. On Recovery Algorithms for Chain Graphs. *International Journal of Approximate Reasoning*, 17:265–293, 1997.

[28] N. Wermuth. On Block-Recursive Linear Regression Equations (with Discussion). *Brazilian Journal of Probability and Statistics*, 6:1–56, 1992.

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

## 5.3 Article 2: Chain graph interpretations and their relations, extended version

### Bibliography:

Title: Chain Graph Interpretations and Their Relations, Extended Version
Authors: Dag Sonntag and Jose M. Peña
Publication: In Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty (2013)
Status: Extended version of accepted paper

# Chain Graph Interpretations and their Relations

Dag Sonntag and Jose M. Peña

ADIT, IDA, Linköping University, Sweden

**Abstract** This paper deals with different chain graph interpretations and the relations between them in terms of representable independence models. Specifically, we study the Lauritzen-Wermuth-Frydenberg, Andersson-Madigan-Perlman and multivariate regression interpretations and present the necessary and sufficient conditions for when a chain graph of one interpretation can be perfectly translated into a chain graph of another interpretation. Moreover, we also present a feasible split for the Andersson-Madigan-Perlman interpretation with similar features as the feasible splits presented for the other two interpretations.
**Keywords:** Chain Graphs, Lauritzen-Wermuth-Frydenberg interpretation, Andersson-Madigan-Perlman interpretation, multivariate regression interpretation.

## 1 Introduction

Today there exist mainly three interpretations of chain graphs (CGs). These are the Lauritzen-Wermuth-Frydenberg (LWF) interpretation presented by Lauritzen, Wermuth and Frydenberg in the late eighties [6, 7], the Andersson-Madigan-Perlman (AMP) interpretation presented by Anderson, Madigan and Perlman in 2001 [2] and the multivariate regression (MVR) interpretation presented by Cox and Wermuth in the nineties [3, 4]. A fourth interpretation of CGs can also be found in a study by Drton [5] but this interpretation has not been further studied and will not be discussed in this paper.

Each interpretation has a different separation criterion and do therefore represent different independence models. So far most papers have studied the different interpretations independently with a few exceptions such as the study of discrete CG models by Drton [5] and the study of CGs representing Gaussian distributions by Wermuth et al. [12]. Therefore it has not really been studied what differences and similarities that exist between the different interpretations in terms of representable independence models. Andersson et al. made a small study of this when they presented their new (AMP) interpretation and managed to show when the independence model of a CG of the AMP interpretation could be represented perfectly by a CG of the LWF interpretation. They did however not show when the opposite held and did no comparison with CGs of the MVR interpretation. Wer-

muth and Sadeghi did on the other hand present conditions for when a CG of the MVR interpretation could be translated into a CG of the LWF or AMP interpretation when they introduced regression graphs [11]. The conditions were however only necessary and sufficient if the two CGs contained the same connectivity components and not the more general case where the CGs could take any form.

In this paper we hope to fill this gap and hence the main contribution of this paper is a table where we show the necessary and sufficient conditions for when a CG of one interpretation can be perfectly translated into a CG of another interpretation. First we do however define a feasible split for the AMP interpretation, with similar features as the feasible splits shown for the LWF [10] and MVR [9] interpretation, that are used in these conditions. Hence this is our second contribution. Finally we also show that there for all three CG interpretations exists a minimal set of non-directed edges for each Markov equivalence class and that the CG containing these, and only these, non-directed edges can be reached through repeated feasible splits from any member of the class.

The remainder of the article is organized as follows. In the next section we present the notation we will use throughout the article. This is followed by the definitions of the feasible splits for each interpretation as well as the proof that the feasible split for CGs of the AMP interpretation is sound. In section 4 we start by presenting the conditions of when a CG of one interpretation can be perfectly represented by a CG of another interpretation. This is then followed by the proofs that these conditions are sound.

# 2 Notation

All graphs are defined over a finite set of variables $V$. If a graph $G$ contains an edge between two nodes $V_1$ and $V_2$, we denote with $V_1 \to V_2$ a *directed edge*, with $V_1 \leftrightarrow V_2$ a *bidirected edge* and with $V_1 - V_2$ an *undirected edge*. By $V_1 \circ\!\!\to V_2$ we mean that either $V_1 \to V_2$ or $V_1 \leftrightarrow V_2$ is in $G$. By $V_1 \circ\!\!- V_2$ we mean that either $V_1 \to V_2$ or $V_1 - V_2$ is in $G$. By $V_1 \circ\!\!-\!\!\circ V_2$ we mean that there exists an edge between $V_1$ and $V_2$ in $G$ while we with $V_1 \cdots V_2$ mean that there might or might not exist an edge between $V_1$ and $V_2$. By a *non-directed* edge we mean either a bidirected edge or an undirected edge. A set of nodes is said to be *complete* if there exist edges between all pairs of nodes in the set.

The *parents* of a set of nodes $X$ of $G$ is the set $pa_G(X) = \{V_1 | V_1 \to V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *children* of $X$ is the set $ch_G(X) = \{V_1 | V_2 \to V_1$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *spouses* of $X$ is the set $sp_G(X) = \{V_1 | V_1 \leftrightarrow V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *neighbours* of $X$ is the set $nb_G(X) = \{V_1 | V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *boundary* of $X$ is the set $bd_G(X) = pa_G(X) \cup nb_G(X) \cup sp_G(X)$. The *adjacents* of $X$ is the set $ad_G(X) = \{V_1 | V_1 \to V_2, V_1 \leftarrow V_2$, $V_1 \leftrightarrow V_2$ or $V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$.

A *route* from a node $V_1$ to a node $V_n$ in $G$ is a sequence of nodes

$V_1, \ldots, V_n$ such that $V_i \in ad_G(V_{i+1})$ for all $1 \leq i < n$. A *path* is a route containing only distinct nodes. The length of a path is the number of edges in the path. A path is called a *cycle* if $V_n = V_1$. A path is *descending* if $V_i \in pa_G(V_{i+1}) \cup sp_G(V_{i+1}) \cup nb_G(V_{i+1})$ for all $1 \leq i < n$. A path $\pi = V_1, \ldots, V_n$ is *minimal* if there exists no other path $\pi_2$ between $V_1$ and $V_n$ st $\pi_2 \subset \pi$ holds. The *descendants* of a set of nodes $X$ of $G$ is the set $de_G(X) = \{V_n|$ there is a descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. A path is *strictly descending* if $V_i \in pa_G(V_{i+1})$ for all $1 \leq i < n$. The *strict descendants* of a set of nodes $X$ of $G$ is the set $sde_G(X) = \{V_n|$ there is a strict descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. The *ancestors* (resp. *strict ancestors*) of $X$ is the set $an_G(X) = \{V_1|V_n \in de_G(V_1), V_1 \notin X, V_n \in X\}$ (resp. $san_G(X) = \{V_1|V_n \in sde_G(V_1), V_1 \notin X, V_n \in X\}$). A cycle is called a *semi-directed cycle* if it is descending and $V_i \rightarrow V_{i+1}$ is in $G$ for some $1 \leq i < n$. A CG under the Lauritzen-Wermuth-Frydenberg (LWF) interpretation, denoted LWF CG, contains only directed and undirected edges but no semi-directed cycles. Likewise a CG under the Andersson-Madigan-Perlman (AMP) interpretation, denoted AMP CG, is a graph containing only directed and undirected edges but no semi-directed cycles. A CG under the multivariate regression (MVR) interpretation, denoted MVR CG, is a graph containing only directed and bidirected edges but no semi-directed cycles. A *connectivity component* $C$ of a LWF CG or an AMP CG (resp. MVR CG) is a maximal (wrt set inclusion) set of nodes such that there exists a path between every pair of nodes in $C$ containing only undirected edges (resp. bidirected edges). We denote the set of all connectivity components in a CG $G$ by $cc(G)$ and the component to which a set of nodes $X$ belong in $G$ by $co_G(X)$. A *subgraph* of $G$ is a subset of nodes and edges in $G$. A subgraph of $G$ induced by a set of its nodes $X$ is the graph over $X$ that has all and only the edges in $G$ whose both ends are in $X$. A *bidirected flag* is an induced subgraph of the form $X \leftrightarrow Y \leftrightarrow Z$ in a MVR CG. With the moral closure graph of a component $C$ in a LWF CG $G$, denoted $(G_{cl(C)})^m$, we mean the subgraph of $G$ induced by $C \cup pa_G(C)$ where every edge have been made undirected and every pair of nodes in $pa_G(C)$ have been made adjacent with undirected edges.

Let $X$, $Y$ and $Z$ denote three disjoint subsets of $V$. We say that $X$ *separated* from $Y$ given $Z$ denoted as $X \perp_G Y|Z$ if the following criteria is met: If $G$ is a LWF CG then $X$ and $Y$ are separated given $Z$ iff there exists no route between $X$ and $Y$ such that every node in a non-collider section on the route is not in $Z$ and some node in every collider section on the route is in $Z$. A *section* of a route is a maximal (wrt set inclusion) non-empty set of nodes $B_1 ... B_n$ such that the route contains the subpath $B_1 - B_2 - \ldots - B_n$. It is called a *collider section* if $B_1 \ldots B_n$ together with the two neighbouring nodes in the route, $A$ and $C$, form the subpath $A \rightarrow B_1 - B_2 - \ldots - B_n \leftarrow C$. For any other configuration the section is a non-collider section. If $G$ is an AMP CG then $X$ and $Y$ is separated given $Z$ iff there exists no S-open path between $X$ and $Y$. A path is said to be *S-open* iff every non-head-no-tail

node on the path is not in $Z$ and every head-no-tail node on the path is in $Z$ or $san_G(Z)$. A node $B$ is said to be a *head-no-tail* in an AMP CG $G$ between two nodes $A$ and $C$ on a path if one of the following configurations exists in $G$: $A \rightarrow B \leftarrow C$, $A \rightarrow B - C$ or $A - B \leftarrow C$. Moreover $G$ is also said to contain a triplex $(\{A, C\}, B)$ iff one such configuration exists in $G$ and $A$ and $C$ are not adjacent in $G$. For any other configuration the node $B$ is a non-collider. If $G$ is a MVR CG then $X$ and $Y$ are separated given $Z$ iff there exists no d-connecting path between $X$ and $Y$. A path is said to be *d-connecting* iff every non-collider on the path is not in $Z$ and every collider on the path is in $Z$ or $san_G(Z)$. A node $B$ is said to be a *collider* in a MVR CG $G$ between two nodes $A$ and $C$ on a path if one of the following configurations exists in $G$: $A \rightarrow B \leftarrow C$, $A \rightarrow B \leftrightarrow C$,$A \leftrightarrow B \leftarrow C$ or $A \leftrightarrow B \leftrightarrow C$. For any other configuration the node $B$ is a non-collider.

The *independence model* $M$ induced by a graph $G$, denoted as $I(G)$ or $I_{PGM-class}(G)$, is the set of separation statements $X \perp_G Y | Z$ that hold in $G$ according to the interpretation to which $G$ belongs or the subscripted PGM-class. We say that two graphs $G$ and $H$ are *Markov equivalent* (under the same interpretation) or that they are in the same *Markov equivalence class* iff $I(G) = I(H)$.

# 3 Feasible splits

For the LWF and MVR interpretation, operations for altering a CG structure without changing its Markov equivalence class have been presented [9, 10]. One such operation is called feasible split and is in this article used to prove certain theorems. Hence we repeat the definitions here. Moreover, we also present the corresponding operation, called feasible split for AMP CGs, for the AMP CG interpretation and prove that it is sound. Note that this is not the inverse operation to a legal merging presented in the deflagging procedure for AMP CGs by Roverto and Studený [8]. Their operation was applied to so called strong equivalence classes, not the more general Markov equivalence classes used here.

**Definition 1.** *Feasible split for LWF CGs [10]*
*A connectivity component $C$ of CG $G$ under the LWF interpretation can be feasibly split into two disjoint sets $U$ and $L$ st $U \cup L = C$ by replacing every undirected edge between $U$ and $L$ with a directed edge orientated towards $L$ iff:*
*1.* $\forall A \in ne_G(L) \cap U, pa_G(L) \subseteq pa_G(A)$
*2.* $ne_G(L) \cap U$ *is complete*

**Definition 2.** *Feasible split for AMP CGs*
*A connectivity component $C$ of CG $G$ under the AMP interpretation can be feasibly split into two disjoint sets $U$ and $L$ st $U \cup L = C$ by replacing every undirected edge between $U$ and $L$ with a directed edge orientated towards $L$ iff:*
*1.* $\forall A \in ne_G(L) \cap U, L \subseteq ne_G(A)$
*2.* $ne_G(L) \cap U$ *is complete*
*3.* $\forall B \in L, pa_G(ne_G(L) \cap U) \subseteq pa_G(B)$

---

**Definition 3.** *Feasible split for MVR CGs [9]*
*A connectivity component $C$ of CG $G$ under the MVR interpretation can be feasible split into two disjoint sets $U$ and $L$ st $U \cup L = C$ by replacing every bidirected edge between $U$ and $L$ with a directed edge orientated towards $L$ iff:*
*1.* $\forall A \in sp_G(U) \cap L$, $U \subseteq sp_G(A)$ *holds*
*2.* $\forall A \in sp_G(U) \cap L$, $pa_G(U) \subseteq pa_G(A)$ *holds*
*3.* $\forall B \in sp_G(L) \cap U$, $sp_G(B) \cap L$ *is a complete set*

---

**Definition 4.** *Maximally orientated CG*
*A CG $G$ (under any interpretation) is maximally orientated iff no feasible splits can be performed on $G$.*

**Lemma 1.** *A CG $G$ of the AMP interpretation is in the same Markov equivalence class before and after a feasible split.*

*Proof.* Assume the contrary. Let $G$ be a CG under the AMP interpretations and $G'$ a graph st $G'$ is $G$ with a feasible split performed upon it. $G$ and $G'$ are in different Markov equivalence classes or $G'$ is not a CG under the AMP interpretation iff (1) $G$ and $G'$ does not have the same adjacencies, (2) $G$ and $G'$ does not have the same triplexes or (3) $G'$ contains semi-directed cycles.

First it is clear that $G$ and $G'$ contains the same adjacencies since a feasible split does not change the adjacencies of any node in $G$. Secondly let us assume $G$ and $G'$ does not have the same triplexes. First let us assume that $G'$ contains a triplex $(\{X, Y\}, Z$ that does not exist in $G$. It is clear that such a triplex can only occur if $Z \in L$ since the only difference between $G$ and $G'$ is that $G'$ contains some directed edges orientated towards $L$ where $G$ contains undirected edges. It is clear that if the triplex is a flag then the one of the node $X$ or $Y$, let's say $X$, must be in $U$ and the other one, let's say $Y$, must be in $L$. However, according to condition 1 $Y$ must be adjacent to $X$ which causes a contradiction. If the triplex is not a flag both $X$ and $Y$ must be in $U$. They also have to be in $ne_G(L)$, which, together with condition 2, contradicts that they are not adjacent. Hence we have a contradiction for that $G'$ contains a triplex that does not exist in $G$.

Secondly assume $G$ contains a triplex $(\{X, Y\}, Z)$ that does not exist in $G'$. It is clear that this new triplex cannot be over a node in $L$ since these nodes only have edges orientated towards them. Instead assume $Z \in U$. This gives that one of the nodes $X$ or $Y$, let's say $X$, must be a parent of $Z$ and the other, let's say $Y$, must be in $L$. This does however contradict condition 3, since every parent of $Z$ also must be a parent of $Y$, and hence $X$ and $Y$ must be adjacent. This gives us a contradiction.

Finally assume $G'$ contain a semi-directed cycle. This means there exists two nodes $X$ and $Y$ st $X \in pa_{G'}(Y)$ but $X \in de_{G'}(Y) \cup co_{G'}(Y)$. It is clear that $\forall A \in V$ $de_{G'}(A) \subseteq de_G(A)$ and $co_{G'}(A) \subseteq co_G(A)$ hold. Hence we must have that $X \in de_G(Y) \cup co_G(Y)$ also hold which, together with $\forall B \in V \setminus L$ $pa_{G'}(B) = pa_G(B)$, means that $Y$ is in $L$ and since $\forall D \in L$ $pa_{G'}(D) = pa_G(D) \cup U$ holds $X$ must be in $U$. However, at the same time $co_{G'}(Y) = co_G Y \setminus U$ and $de_{G'}(Y) \subseteq de_G Y$ must hold and hence we have a contradiction. □

A maximally orientated CG can be obtained from any member of its Markov equivalence class by performing feasible splits until no more feasible splits can be performed.

**Theorem 1.** *A CG (under any interpretation) has the minimal set of non-directed edges for its Markov equivalence class if no feasible split is possible.*

The following theorem shows that there may exist several maximally orientated CGs in a given Markov equivalence class but all of them share the same non-directed edges.

**Theorem 2.** *For any Markov equivalence class of CGs (under any interpretation), there exists a unique minimal (wrt inclusion) set of non-directed edges that is shared by all members of the class.*

The proofs of the Theorem 1 and 2 for the MVR interpretation can be found in the article by Sonntag and Peña [9]. These proofs can easily be adapted for the LWF and AMP interpretations.

# 4 Translations between interpretations

In this section the main result of this paper is presented, namely what the conditions are for a CG of one interpretation to be possible to translate into a CG of another interpretation. With translate we mean that the induced independence model of a CG of one interpretation can be represented perfectly by a CG of another interpretation. A summary of these results is presented in Table 1.

From the table two things can be noted. First that the conditions given in the table may include a maximally oriented CG $G'$ in the same equivalence class as $G$. This is done for several reasons. First, such a graph is easy and

| | LWF | AMP | MVR |
|---|---|---|---|
| LWF | - | Unidentified | $(G_{cl(K)})^m$ is chordal for all $K \in cc(G)$. |
| AMP | $G$ contains no $k$-biflag where $k \geq 2$ [2] | - | $G'$ does not contain any induced subgraph of the form $X$–$Y$–$Z$ |
| MVR | $G'$ contains no bidirected edge | $G'$ contains no bidirected flag | - |

Table 1: Given a CG $G$ of the interpretation denoted in the row, and a maximally oriented CG $G'$ in the Markov equivalence class of $G$, there exists a CG $H$ of the interpretation denoted in the column st $G$ and $H$ are Markov equivalent iff the condition in the intersecting cell is fulfilled.

computationally simple to find. Secondly, this allows the proofs to be based on the idea that no feasible split is possible for the interpretation in mind. Third and last, the search space of CGs is smaller and more assumptions can be made on the CG. This in turn allows for more efficient algorithms when calculating if the condition holds for some CG. The second note that can be made is that there still does not exist any necessary and sufficient condition for when a perfect translation of a LWF CG $G$ into an AMP CG $H$ is possible. Andersson et al. gave a necessary condition but also showed that this condition was not sufficient [2]. We have managed to prove the necessity of more elaborate conditions but still been unable to prove sufficiency for these. Hence this condition is left for future work.

The rest of this section contains the theorems stating the conditions shown in Table 1 together with their proofs.

## 4.1 Translation of MVR CGs to AMP CGs

**Theorem 3.** *Given a MVR CG $G$, and a maximally oriented MVR CG $G'$ in the Markov equivalence class of $G$, there exists an AMP CG $H$ st $I_{MVR}(G) = I_{AMP}(H)$ iff $G'$ contains no bidirected flag.*

*Proof.* Sufficiency follows from from Lemmas 4 and 5 and necessity follows from Lemma 2. □

**Lemma 2.** *A MVR CG $G$ and an AMP CG $H$ with the same structure, except that every bidirected edge in $G$ is replaced by a undirected edge in $H$ and where $G$ contains no bidirected flag, represent the same independence model.*

*Proof.* Assume to contrary that there exists two CGs, $G$ under the MVR interpretation and $H$ under the AMP interpretation, st $G$ does not contain any bidirected flag, i.e induced subgraph of the form $X \leftrightarrow Y \leftrightarrow Z$, $G$ and $H$ contain the same directed edges, and for every bidirected edge in $G$ $H$ has an undirected edge instead (and only contains those undirected edges) but

$I_{MVR}(G) \neq I_{AMP}(H)$. Clearly we must have $V_G = V_H$ and that $adj_G(X) = adj_H(X), pa_G(X) = pa_H(X)$ and $co_G(X) = co_H(X)$ holds for all $X \in V_G$. Given the definition of strict descendants $san_G(X) = san_H(X)$ must also hold. Moreover note that $H$ cannot contain any induced subgraph of the form $X - Y - Z$. Finally note that both $G$ and $H$ contains the same paths between $X$ and $Y$.

For $I(G) \neq I(H)$ to hold there has to exist a path $\pi$ in $G$ (resp. $H$) that is d-connecting (resp. S-open) st there exist no path in $H$ (resp. $G$) that is S-open (resp. d-connecting). Let $\pi$ be a minimal d-connecting (resp. S-open) path in $G$ (resp. $H$). Note that $\pi$ cannot contain any contain any subpath of the form $V_1 \leftrightarrow V_2 \leftrightarrow V_3$ (resp. $V_1 - V_2 - V_3$) since the edge $V_1 \leftrightarrow V_3$ (resp. $V_1 - V_3$) must exist in $G$ (resp. $H$) or $G$ contains a bidirected flag or semi-directed cycle. This in turn would mean that $\pi$ is not minimal since the path $\pi \setminus V_2$ also must be d-connecting and shorter than $\pi$. For $\pi$ to be both d-connecting and S-open for any set of nodes $Z$ it must contain the same colliders and head-no-tail nodes. A node $W \in \pi$ is a collider if it is part of the following configurations of edges in $\pi$ (1) $\rightarrow W \leftarrow$, (2) $\leftrightarrow W \leftarrow$, (3) $\rightarrow W \leftrightarrow$ and (4) $\leftrightarrow W \leftrightarrow$. Clearly the fourth case cannot occur. Case 1-3 would be translated into (1) $\rightarrow W \leftarrow$, (2) $- W \leftarrow$, (3)$\rightarrow W -$ in $H$ which are all (and the only) head-no-tail configurations. Hence $\pi$ must be d-connecting in $G$ iff $\pi$ is S-open in $H$ which contradicts the assumption. □

**Lemma 3.** *If a maximally oriented CG $G$ of the MVR interpretation contains a bidirected flag $X \leftrightarrow Y \leftrightarrow Z$ then $G$ also contains an induced subgraph of the form shown in (1) Figure 1a or (2) 1b or (3) $P \circ\!\!\rightarrow Q \leftrightarrow Y \leftrightarrow Z$ or (4) $P \circ\!\!\rightarrow Q \leftrightarrow W \leftrightarrow Z$ st $bd_G(Q) \subseteq bd_G(Y) \cup Y$ and $Y \in sp_G(Q)$ hold.*

*Proof.* Assume the contrary, that no such induced subgraph exists in $G$ even though $G$ contains a bidirected flag and $G$ is maximally orientated. Let $C$ be the component of which $X, Y$ and $Z$ belongs. Let $A$ be the set of nodes $A_k$ st $A_k \in sp_G(Y)$ but $A_k \notin sp_G(Z)$. We know that $X$ fulfills these criteria and hence $|A| \geq 1$.

First note that if there exists a node $A_k \in A$ st $bd_G(A_k) \notin bd_G(Y) \cup Y$ then there exists an induced subgraph $P \circ\!\!\rightarrow A_k \leftrightarrow Y \leftrightarrow Z \cdots P$ in $G$ for some node $P \in bd_G(A_k) \setminus bd_G(Y) \setminus Y$. Hence we have a contradiction since $G$ either

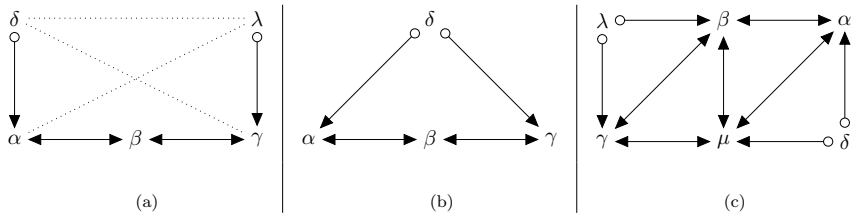



Figure 1: MVR subgraph forms

contains an induced subgraph of the form shown in Figure 1b ($P \in bd_G(Z)$) or of the form $P \circ\!\!\rightarrow Q \leftrightarrow Y \leftrightarrow Z$ ($P \notin bd_G(Z)$). Therefore we must have that $bd_G(A_k) \subseteq Y \cup bd_G(Y)$ holds for all $A_k \in A$, i.e. that $bd_G(A) \subseteq Y \cup bd_G(Y)$ holds.

Secondly note that we can let $B$ be a subset of $A$ st $B$ consists of the nodes in one connected subgraph in the subgraph of $G$ induced by $A$ (any connected subgraph will do). Let $D$ be the set of nodes st $D = sp_G(Y) \cap sp_G(Z) \cap sp_G(B)$. With these sets we know that the spouses of $Y$ can be either adjacent of $Z$ or not, hence $sp_G(Y) = D \cup A$ must hold. This in turn gives that $sp_G(A) = D \cup Y$ and $bd_G(A) \subseteq D \cup Y \cup pa_G(Y)$ since $\forall A_k \in A\ bd_G(A_k) \subseteq Y \cup bd_G(Y)$ holds. Moreover $sp_G(B) = D \cup Y$ and $bd_G(B) \subseteq D \cup Y \cup pa_G(Y)$ must also hold. Hence, if $D$ is empty then $sp_G(B) = \{Y\}$ and $bd_G(B) \subseteq Y \cup pa_G(Y)$ must hold. This does however lead to a contradiction because a split then is possible st $U$ consists of $B$ and $L$ consists of $C \setminus U$. Hence there has to exists at least one node in $D$.

Thirdly note that $D \cup Y$ must be complete or the induced subpath $B_k \leftrightarrow DY_i \leftrightarrow Z \leftrightarrow DY_j \leftrightarrow B_1 \leftrightarrow ... \leftrightarrow B_l \leftrightarrow B_k$, $l \geq 0$, exists in $G$ for some nodes $B_k, B_1, ..., B_l \in B$ and $DY_i, DY_j \in D \cup Y$. This means that $G$ contains an induced subgraph of the form shown in either Figure 1a ($l > 0$) or 1b ($l = 0$).

Fourth and finally note that there must exist a node $P$ st $P \in bd_G(B) \cup B$ but $P \notin bd_G(D_j)$ for some $D_j \in D \cup Y$ or a split is feasible where $U$ consists of $B$ and $L$ consists of $C \setminus U$. Note that $D_j \neq Y$ must hold since $bd_G(B) \cup B \subseteq bd_G(Y) \cup Y$. This means that there must exist 2 nodes $B_i, D_j$ st $P \in bd_G(B_i), P \notin bd_G(D_j), B_i \in B$, $B_i \in sp(D_j)$ and $D_j \in D$ st the induced subgraph $P \circ\!\!\rightarrow B_i \leftrightarrow D_j \leftrightarrow Z \cdots P$ exist in $G$. This is a contradiction either because $G$ contains an induced subgraph of the form shown in Figure 1b ($P \in bd_G(Z)$) or $P \circ\!\!\rightarrow B_i \leftrightarrow D_j \leftrightarrow Z$ ($P \notin bd_G(Z)$) where $bd_G(B_i) \subseteq bd_G(Y) \cup Y$ and $Y \in sp_G(B_i)$ holds. □

**Lemma 4.** *If a maximally oriented CG $G$ of the MVR interpretation contains a bidirected flag then $G$ at least one of the induced subgraphs shown in Figure 1 exists in $G$.*

*Proof.* Assume the contrary, that no such induced subgraph exists in $G$ even though $G$ contains a bidirected flag and $G$ is maximally orientated. Since $G$ contains a bidirected flag we do with Lemma 3 get that $G$ must contain an induced subgraph $X \leftrightarrow Y \leftrightarrow Z \leftarrow\!\circ W$ or a contradiction directly follows. If we now apply Lemma 3 to $X \leftrightarrow Y \leftrightarrow Z$ we get that, since for $G$ to contain any induced subgraph of the form shown in Figure 1a or 1b is a contradiction, there exist a set of nodes (that can be renamed to) $c_1, c_2, c_3$ st the induced subgraph $c_1 \circ\!\!\rightarrow c_2 \leftrightarrow c_3 \leftrightarrow Z$ exists in $G$ and $c_3 = Y$ holds or $bd_G(c_2) \subseteq bd_G(Y) \cup Y$ and $Y \in sp_G(c_2)$ hold. If $c_3 = Y$, $G$ must contain the subgraph $c_1 \circ\!\!\rightarrow c_2 \leftrightarrow Y \leftrightarrow Z \leftarrow\!\circ W$ where $c_1 \notin adj_G(Y)$ and $W \notin adj_G(Y)$ must hold and $c_1 = W$ might hold. Clearly this subgraph takes the form of either Figure 1a ($c_1 \neq W$) or 1b ($c_1 = W$) which is a contradiction. Hence $c_3 \neq Y$, $bd_G(c_2) \subseteq bd_G(Y) \cup Y$ and $Y \in sp_G(c_2)$ must hold.

Since $W \notin adj_G(Y)$ holds and $bd_G(c_2) \subseteq bd_G(Y) \cup Y$ it is clear that $c_1, c_3 \in bd_G(Y)$ must hold. Hence $W \neq c_2$ holds since $W \notin adj_G(Y) \cup Y$. This in turn means that $W \notin bd_G(c_2)$ holds since $bd_G(c_2) \subseteq bd_G(Y) \cup Y$ and $W \notin bd_G(Y) \cup Y$. Finally we can see that $W \in bd_G(c_3)$ holds or the induced subgraph $c_1 \multimap c_2 \leftrightarrow c_3 \leftrightarrow Z \leftarrow\!\circ W$ takes the form shown in Figure 1a ($c_1 \neq W$) or 1b ($c_1 = W$). However, if $W \in bd_G(c_3)$ holds $G$ contains an induced subgraph of the form shown in Figure 1c (where $\delta = W$, $\lambda = c_1$, $\mu = c_3$, $\gamma = c_2$, $\beta = Y$ and $\alpha = Z$) and we have a a contradiction. □

**Lemma 5.** *The independence model of a CG $G$ of the MVR interpretation which contains an induced subgraph of one of the forms shown in Figure 1 cannot be perfectly represented as a CG $H$ of the AMP interpretation.*

*Proof.* Assume the contrary, that there exists a CG $H$ under the AMP interpretation that can represent these independence models.

First assume that the independence model of the graph shown in Figure 1a can be represented in a CG $H$ of the AMP interpretation. It is clear that $H$ must have the same skeleton, or clearly some separations or non-separations that hold in $G$ would not hold in $H$. The following independence statements holds in $G$: $\delta \perp_G \beta | pa_G(\beta)$, $\alpha \perp_G \gamma | pa_G(\alpha)$ and $\beta \perp_G \lambda | pa_G(\beta)$. $\delta \perp_G \beta | pa_G(\beta)$ gives us that a triplex $(\{\delta, \beta\}, \alpha)$ must exist in $H$, since $\alpha \notin pa_G(\beta)$ i.e. that (1) $\delta \rightarrow \alpha - \beta$, (2) $\delta - \alpha \leftarrow \beta$ or (3) $\delta \rightarrow \alpha \leftarrow \beta$ exists in $H$. $\alpha \perp_G \gamma | pa_G(\alpha)$ does however also state that a triplex $(\{\alpha, \gamma\}, \beta)$ must exist in $H$, since $\beta \notin pa_G(\alpha)$. For this to happen the edge between $\alpha$ and $\beta$ cannot be orientated towards $\alpha$ hence the subgraph $\delta \rightarrow \alpha - \beta \leftarrow \gamma$ must exist in $H$. The orientation of the edge between $\beta$ and $\gamma$ does however contradict the third independence statement $\beta \perp_G \lambda | pa_G(\beta)$ which implies that the triplex $(\{\beta, \lambda\}, \gamma)$ must exist in $H$, since $\gamma \notin pa_G(\beta)$. Hence we have a contradiction if $G$ contains the induced subgraph shown in Figure 1a.

Secondly assume that the independence model of the graph shown in Figure 1b can be represented in a CG $H$ of the AMP interpretation. It is clear that $H$ must have the same skeleton, or clearly some separations or non-separations that hold in $G$ would not hold in $H$. The following independence statements must then hold in $G$: $\delta \perp_G \beta | pa_G(\beta)$ and $\alpha \perp_G \gamma | pa_G(\alpha)$. $\delta \perp_G \beta | pa_G(\beta)$ gives us that two triplexes must exist in $H$, first $(\{\delta, \beta\}, \alpha)$ and secondly $(\{\delta, \beta\}, \gamma)$, since $\alpha, \gamma \notin pa_G(\beta)$. $(\{\delta, \beta\}, \alpha)$ gives that one of the following configurations must occure in $H$: (1) $\delta - \alpha \leftarrow \beta$, (2) $\delta \rightarrow \alpha - \beta$ or (3) $\delta \rightarrow \alpha \leftarrow \beta$. However, the independence statement $\alpha \perp_G \gamma | pa_G(\alpha)$ implies that the triplex $(\{\alpha, \gamma\}, \beta)$ must exist in $H$ since $\beta \notin pa_G(\alpha)$. If the triplex $(\{\alpha, \gamma\}, \beta)$ should hold in $H$ the edge between $\alpha$ and $\beta$ cannot be orientated towards $\alpha$ hence the subgraph $\delta \rightarrow \alpha - \beta \leftarrow \gamma$ must exist in $H$. The orientation of the edge between $\beta$ and $\gamma$ does however contradict the triplex $(\{\delta, \beta\}, \gamma)$ and hence we have a contradiction for the $G$ shown in Figure 1b.

Third and last assume that the independence model of the graph shown in Figure 1c can be represented in a CG $H$ of the AMP interpretation. From the Figure we can read the following independence statements: $\lambda \perp_G \mu | pa_G(\mu)$,

$\alpha \perp_G \gamma | pa_G(\alpha), \beta \perp_G \delta | pa_G(\beta)$. It is clear that $H$ must have the same skeleton, or clearly some separations or non-separations that hold in $G$ would not hold in $H$. $\lambda \perp_G \mu | pa_G(\mu)$ and $\alpha \perp_G \gamma | pa_G(\alpha)$ gives that the triplexes $(\{\lambda, \mu\}, \beta)$ and $(\{\alpha, \gamma\}, \mu)$ must exists in $H$ since $\beta \notin pa_G(\mu)$ and $\mu \notin pa_G(\alpha)$. As seen above this gives that $\lambda \rightarrow \gamma - \mu \leftarrow \alpha$ must exist in $H$. Similarly $\beta \perp_G \delta | pa_G(\beta)$ and $\lambda \perp_G \mu | pa_G(\mu)$ gives that $\lambda \rightarrow \beta - \mu \leftarrow \delta$ must exist in $H$. Finally $\alpha \perp_G \gamma | pa_G(\alpha)$ and $\beta \perp_G \delta | pa_G(\beta)$ gives that the triplexes $(\{\alpha, \gamma\}, \beta)$ and $(\{\beta, \delta\}, \alpha)$ must hold in $H$, since $\beta \notin pa_G(\alpha)$ and $\alpha \notin pa_G(\beta)$, which in turn gives that $\gamma \rightarrow \beta - \alpha \leftarrow \delta$ must exist in $H$. This does however contradict that $H$ is a CG since the semi-directed cycle $\gamma \rightarrow \beta - \mu - \gamma$ exists in $H$. Hence we have a contradiction. □

## 4.2 Translation of AMP CGs to MVR CGs

**Theorem 4.** *Given an AMP CG $G$, and a maximally oriented AMP CG $G'$ in the Markov equivalence class of $G$, there exists a CG $H$ st $I_{AMP}(G) = I_{MVR}(H)$ iff $G'$ does not contain any induced subgraph of the form $X - Y - Z$.*

*Proof.* Sufficiency follows from Lemma 2 while necessity follows from 6. □

**Lemma 6.** *If a maximally orientated CG $G$ of the AMP interpretation contains an induced subgraph of the form $X - Y - Z$ then $G$ there exists no CG $H$ of the MVR interpretation st $I_{AMP}(G) = I_{MVR}(H)$.*

*Proof.* Assume to the contrary that the lemma does not hold. Clearly $G$ and $H$ must have the same skeleton or some separations in $H$ do not hold in $G$ or vice versa. Let $H$ have a component ordering $ord$ for its components $c_1, ...c_k$ st $ord(c_i) < ord(c_j)$ if $c_i$ is a parent of $c_j$. Let $C$ be the component of $X$ in $G$. From the assumption we know that $X \perp_G Z | nb_G(X) \cup pa_G(X \cup nb_G(X))$ holds, where $Y \in nb_G(X)$, and hence that $H$ must contain one of the following induced subgraphs: $X \circ\!\!\rightarrow Y \rightarrow Z$, $X \leftarrow Y \leftrightarrow Z$ or $X \leftarrow Y \rightarrow Z$. For any other configuration of edges $X \perp_H Z | nb_G(X) \cup pa_G(X \cup nb_G(X))$ does not hold. Moreover we can generalize the configurations to $X \circ\!\!\rightarrow Y \rightarrow Z$ and $X \leftarrow Y \rightarrow Z$ simply by choosing the nodes to represent $X$ and $Z$ accordingly. Both these structures are included in $X \circ\!\!-\!\!\circ Y \rightarrow Z$ and we will now show that this structure leads to a contradiction if a split is not feasible in $G$.

The proof is iterative and when a restart is noted this is where the proof restarts. For each restart it will be shown that there must exist a triplet of nodes $X$, $Y$, $Z$ st an induced subgraph of the form $X - Y - Z$ exists in $G$ and $X \circ\!\!-\!\!\circ Y \rightarrow Z$ in $H$. Apart from this we also know that $I_{AMP}(G) = I_{MVR}(H)$ holds and that no split is feasible in $G$. Let the set $U$ consist of $Y$ and every node connected by a path to $Y$ in the subgraph of $G$ induced by $C \setminus Z$ and the set $L$ consist of $C \setminus U$. This separation of sets gives that $nb_G(U) = Z$. For a split not to be feasible with these sets one of the conditions in Definition 2 must fail:

Case 1, condition 1 or 2 fails. This means that there exist two nodes $W \in C$ and $P \in C$ st the induced subgraph $P - Z - W$ exists in $G$. Note that one

of the nodes might be $Y$. This means that $P \perp_G W | nb_G(W) \cup pa_G(W \cup nb_G(W))$ holds, where $Z \in nb_G(W)$ and hence that $P \circ\!\!\rightarrow Z \rightarrow W$, $P \leftarrow Z \leftarrow\!\!\circ W$ or $P \leftarrow Z \rightarrow W$ must exist in $H$ as described above. Without losing generality we can say that either $P \circ\!\!\rightarrow Z \rightarrow W$ or $P \leftarrow Z \rightarrow W$ exists in $H$ and that $W \neq Y$ by choosing $P$ and $W$ appropriately. This means that we can restart the proof with the structure $P \circ\!\!-\!\!\circ Z \rightarrow W$ in $H$ (and $P - Z - W$ in $G$). The number of restarts is bounded since (1) the number of nodes in $V$ is bounded and that $ord(co_H(Z)) > ord(co_H(Y))$.

Case 2, condition 1 and 2 hold but condition 3 fails. This means that there exists two nodes $W \in U$ and $P \notin C$ st the induced subgraph $Z - W \leftarrow P$ exists in $G$. First let us cover the case where $W = Y$. This means that $Z \perp_G P | pa_G(Z)$ holds. Since $H$ have the same skeleton as $G$ this means that $H$ must contain an induced subgraph of the form $P \circ\!\!\rightarrow Y \leftarrow\!\!\circ Z$ since $Y \notin pa_G(Z)$. At the same time we know that $H$ contains the edge $Y \rightarrow Z$ which causes a contradiction and hence $Y \neq W$ must hold. Therefore, $P \notin pa_G(Y)$ holds which generalized means that that $pa_G(Y) \subseteq pa_G(Z)$ must hold. For $Z \perp_G P | pa_G(Z)$ to hold in $H$ there must exist an unshielded collider between $Z$ and $P$ over $W$ and hence that the induced subgraph $Z \circ\!\!\rightarrow W \leftarrow\!\!\circ P$ exists in $H$. Similarly we have that $Y \perp_G P | pa_G(Y)$ gives that $H$ contains an induced subgraph of the form $Y \circ\!\!\rightarrow W \leftarrow\!\!\circ P$. Note that $Y \in adj_G(W)$ must hold since condition 2 holds. Moreover for $H$ not to contain a semi directed cycle over $Y \rightarrow Z \circ\!\!\rightarrow W \leftarrow\!\!\circ Y$ we can see that $Y \rightarrow W \leftarrow\!\!\circ P$ must exist in $H$. Finally note that $X \neq W$ must hold since $X \notin adj_G(Z)$ holds.

Now assume $X \in nb_G(W)$. For $X \perp_H Z | nb_G(X) \cup pa_G(X \cup nb_G(X))$ to hold, together with $W \in nb_G(X)$ and $Z \circ\!\!\rightarrow W$, it is easy to see that the induced subgraph $Z \circ\!\!\rightarrow W \rightarrow X$ must be in $H$. We can now see that $P \in pa_G(X)$ must hold or the induced subgraph $X \leftarrow W \leftarrow\!\!\circ P$ in $H$ contradicts that $X \perp_G P | pa_G(X)$ holds in $H$. Moreover, for $X \circ\!\!-\!\!\circ Y \rightarrow W \rightarrow X$ not to form a semi-directed cycle in $H$ the edge between $X$ and $Y$ must be orientated to $X \leftarrow Y$. We can therefore restart the proof by replacing $X$ with $Z$, i.e. with the induced subgraph $X \leftarrow Y \circ\!\!-\!\!\circ Z$ in $H$ (and $X - Y - Z$) in $G$. Since we know that $Z \circ\!\!\rightarrow W \rightarrow X$ exists in $H$ we know that $ord(co_H(X)) > ord(co_H(Z))$. Hence we cannot get back to this subcase again (or we would have that $ord(co_H(X)) < ord(co_H(Z))$ which is a contradiction). This, together with that $Y$ is kept the same and that $|V|$ is finite gives that the number of restarts is bounded. Hence $X \notin nb_G(W)$ must hold.

Now assume that $pa_G(Z) \subseteq pa_G(W)$. We can now restart the proof with $X \circ\!\!-\!\!\circ Y \rightarrow W$. The number of iterations is then bounded since $|V|$ is finite and case 2 cannot occur with $Z$ as $W$ again, or $pa_G(Z) \notin pa_G(W)$ would have to hold which is a contradiction. Hence $pa_G(Z) \notin pa_G(W)$ must hold. Let $Q$ be the parent of $Z$ not shared by $W$. Since $W \perp_G Q | pa_G(W)$ holds, and we know that $H$ contains the induced subgraph $Q \circ\!\!-\!\!\circ Z \circ\!\!\rightarrow W \leftarrow\!\!\circ P$, we can draw the conclusion that $H$ must contain the induced subgraph $Q \circ\!\!\rightarrow Z \leftrightarrow W \leftarrow\!\!\circ P$ since $Z \notin pa_G(W)$. Note that if there exist two different nodes $W_1$ and $W_2$ such that both have the properties described for $W$ in case 2 $W_1$ and

$W_2$ must be adjacent. If this were not the case we would have that both $W_1 \perp_G W_2 | nb_G(W_1) \cup pa_G(W_1 \cup nb_G(W_1))$ and $W_1 \not\perp_H W_2 | nb_G(W_1) \cup pa_G(W_1 \cup nb_G(W_1))$ would hold, since $Z \in nb_G(W_1)$. Also note that since $W_1 \leftrightarrow Z$ and $W_2 \leftrightarrow Z$ exists in $H$ and the edge between $W_1$ and $W_2$ must be bidirected or $H$ contains a semi-directed cycle. Let $D$ be a set of nodes containg $Z$ as well as all nodes that have the properties described for $W$. From the description above we can see that $D$ must be complete and that the subgraph induced by $D$ in $H$ must only contain bidirected edges. We will now show that a split must be feasible in $G$ with $D$ as $L$ and $C \setminus D$ as $U$. For a split not to be feasible one of the constraints in Definition 2 must fail.

Assume condition (1) or (2) fails. Then there exists three nodes $R \in C$, $T \in C$ and $D_j \in D$ st the induced subgraph $T - D_j - R$ exists in $G$. Since $T \perp_G R | nb_G(R) \cup pa_G(R \cup nb_G(R))$ holds we must, without losing generalization, have that $H$ contains the induced subgraph $T \circ\!\!-\!\!\circ D_j \rightarrow R$, since $D_j \in nb_G(R)$. If this is the case we can however restart the proof with this induced subgraph and know that the number of iterations is bounded since $|V|$ is finite and $ord(co_H(D_j)) > ord(co_H(Y))$.

Assume condition (1) and (2) holds but (3) fails. Then there exists two nodes $R \in U$ and $T \notin C$ st the induced subgraph $D_i - R \leftarrow T$ exists in $G$ for some $D_i \in D$. First note that $R$ must be adjacent of all nodes in $D$ or condition 1 would have failed in this split. Secondly note that $R - Y$ must exist in $G$ or condition 2 would fail if we restart the proof with $X \circ\!\!-\!\!\circ Y \rightarrow D_i$ and a contradiction follows from there. Thirdly note that $R \notin adj_G(X)$ must hold or the proof could be restarted with $X \circ\!\!-\!\!\circ Y \rightarrow D_i$, for which condition 3 would fail with $R$ as $W$ and a contradiction would follow as shown above. Finally note that $pa_G(R) \subseteq pa_G(Z)$ must hold or $R$ would be in $D$. This means that $pa_G(W) \nsubseteq pa_G(R)$, and hence that $P \notin pa_G(R)$, holds. Moreover we know that the edge $D_i \leftrightarrow W$ exists in $G$. For $D_i \perp_G T | pa_G(D_i)$ to hold in $H$ it is clear that $H$ must contain the induced subgraph $D_i \circ\!\!\rightarrow R \leftarrow\!\!\circ T$ since $R \notin pa_G(D_i)$. Similarly we have that for $R \perp_H P | pa_G(R)$ to hold $H$ must contain an induced subgraph of the form $R \circ\!\!\rightarrow W \leftarrow\!\!\circ P$ since $W \notin pa_G(R)$. This means that for $R \circ\!\!\rightarrow W \leftrightarrow D_i \circ\!\!\rightarrow R$ not to form a semi-directed cycle in $H$ the edge $R \leftrightarrow D_i$ must exist in $H$. Moreover, since $\forall D_m \in D \setminus D_i \; R \in adj_G(D_m)$, $R \leftrightarrow D_i$ and $D_i \leftrightarrow D_m$ hold, clearly $R \leftrightarrow D_m$ must also hold or $G$ contains a semi-directed cycle. Hence the subgraph of $H$ induced by $D \cup R$ is complete and contains only bidirected edges. This in turn means that for $Y \rightarrow D_i \leftrightarrow R \circ\!\!-\!\!\circ Y$ to not form a semi-directed cycle $Y \rightarrow R$ must exist in $H$. Hence we can move $R$ into $D$ and redo do the last split again. The number of restarts are bounded since $|V|$ is finite.

Hence each condition in Definition 2 must hold and we have a contradiction. □

## 4.3 Translation of MVR CGs to LWF CGs

**Theorem 5.** *Given a MVR CG $G$, and a maximally oriented MVR CG $G'$ that is in the same Markov equivalence class as $G$, there exist a LWF CG $H$ st $I_{MVR}(G) = I_{LWF}(H)$ iff $G'$ contains no bidirected edge, i.e. can be represented as a BN.*

*Proof.* From Lemma 7 it follows that a maximally oriented CG $G'$ of the MVR interpretation with a bidirected edge must have a subgraph of the form shown in Figure 2. If it does not contain any bidirected edge in the maximally oriented model it trivially follows that it is a BN (and hence it can be represented as a CG of the LWF interpretation). From Lemma 8 it then follows that no CG $G$ of the MVR interpretation which contains a subgraph of the form shown in Figure 2 can be represented as a CG of the LWF interpretation. □

**Lemma 7.** *If a bidirected edge exists in a maximally oriented CG $G$ of the MVR interpretation then $G$ must contain an induced subgraph of the form shown in Figure 2.*

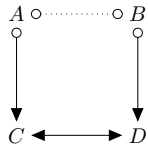



Figure 2: Included subgraph in Lemma 7 and 8.

*Proof.* Assume to the contrary that a CG $G$ of the MVR interpretation exists where (1) no induced subgraph of the form shown in Figure 2 exists, (2) no split is feasible and (3) at least one bidirected edge exists. From this assumption we can see that there has to exist at least two nodes $X$ and $Y$ st $X \leftrightarrow Y$ exists in $G$. Let $C$ be the connectivity component to which $X$ and $Y$ belongs. Separate the nodes of $C$ into two sets $U$ and $L$ st $X$ and every node connected by a path to $X$ in the subgraph of $G$ induced by $C \setminus Y$ belongs to $L$ and $C \setminus L$ belongs to $U$. This separation of nodes allows us to know that $sp_G(L)$ only contains $Y$. For a split not to be feasible at least one condition in Definition 3 has to fail.

Case 1. Assume constraint 1 fails. This means a node $Z \in L$ exist st $Z \leftrightarrow Y \leftrightarrow X$ occurs in $G$ where $Z \notin adj_G(X)$ must hold, or $Z$ would be in $U$. Now remove $Y$ from $U$ and add it to $L$ as well as all nodes not connected by a path with $Z$ in the subgraph of $G$ induced by $U \setminus Y$ and attempt another split. This separation of nodes allows us, since we previously had $nb_G(L) = Y$ and $Y$ now have changed sets, to say that $sp_G(U) = Y$ must hold and hence that constraint 3 cannot fail. However, if constraint 1 or 2

fails we know there exists a node $W$ st $W \circ\!\!\rightarrow Z \leftrightarrow Y \leftrightarrow X$ is a subgraph of $G$ but where $W \notin adj_G(Y)$, and $Z \notin adj_G(X)$ and $W \notin adj_G(X)$ by definition of the initial split, which implies a contradictory induced subgraph. Hence constraint 1 cannot fail in the initial split.

Case 2. If constraint 2 or 3 fails in the initial split we know there exists two nodes $V_1$ and $P_1$ st $P_1 \circ\!\!\rightarrow Y \leftrightarrow V_1$ exists in $G$ but where $V_1 \notin adj_G(P_1)$ (note that $V_1$ might be $X$). Now let $L$ consist of every node connected by a path to $Y$ in the subgraph of $G$ induced by $C \setminus V_1$ and the nodes $C \setminus L$ belong to $U$. This separation of nodes allows us to know that $sp_G(L)$ only contains $V_1$. If constraint 1 fails when performing a split with these sets it is clear from case 1 that a contradiction occurs. If constraint 2 or 3 fails we know there exists two new nodes $V_2$ and $P_2$ st $P_2 \circ\!\!\rightarrow V_1 \leftrightarrow V_2$ exists in $G$ but where $P_2 \notin adj_G(V_2)$. Note that $V_2$ or $P_2$ cannot be $P_1$ since $P_1 \notin adj_G(P_1)$. We now get that $V_2$ cannot be $Y$ or an induced subgraph like that in Figure 2 occurs. $V_2 \in adj(Y)$ and $P_2 \in adj(Y)$ must also hold or the induced subgraph $V_2$ (resp. $P_2$)$\circ\!\!\rightarrow V_1 \leftrightarrow Y \leftarrow\!\!\circ P_1$ occurs. By replacing $V_1$ with $V_2$, setting the proper $U$ and $L$ as described above it we can now repeat this procedure iteratively. Moreover, for every repetition $i$ we must have that $V_i$ and $P_i$ must be adjacent of every $V_j (j < i)$ as well as $Y$ or a contradiction occurs. This means that any nodes $V_i$ and $P_i$ already used in a previous repetition cannot be used in a later one, or both $P_i \in adj(V_i)$ and $P_i \notin adj(V_i)$ would have to hold. This in turn means that the number of repetitions is bounded since $|C|$ is finite and hence we have a contradiction that condition 2 or 3 can fail.

This means that all three conditions in Definition 3 must hold and hence a split must be feasible if the induced subgraph shown in Figure 2 does not occur. □

**Lemma 8.** *If a CG $G$ of the MVR interpretation contains an induced subgraph of the form shown in Figure 2 then $G$ cannot be translated into a CG $H$ of the LWF interpretation.*

*Proof.* Assume to the contrary that there exists a CG $H$, of the LWF interpretation, with the same independence model as $G$ while $G$ contains an induced subgraph of the form shown in Figure 2. Clearly $H$ and $G$ must contain the same nodes and adjacencies or some separations or non-separations must exist in $G$ but not in $H$.

From Figure 2 we can read that $A \perp_G D | pa_G(D)$ and $C \perp_G B | pa_G(C)$ hold. For $A \perp_G D | pa_G(D)$ to hold in $H$ $C$ must be a collider between $A$ and $D$ and hence $H$ must contain the induced subgraph $A \rightarrow C \leftarrow D$. Similarly $C \perp_G B | pa_G(C)$ gives that $H$ must contain the induced subgraph $C \rightarrow D \leftarrow B$ and hence we have a contradiction. □

### 4.4 Translation of LWF CGs to MVR CGs

**Theorem 6.** *Given a LWF CG $G$ there exists a CG $H$ st $I_{LWF}(G) = I_{MVR}(H)$ iff $(G_{cl(K)})^m$ is chordal for all $K \in cc(G)$.*

*Proof.* To prove the "if" part, note that if $(G_{cl(K)})^m$ is chordal for all $K \in cc(G)$, then there is a DAG $D$ st $I_{LWF}(G) = I_{BN}(D)$ [1, Proposition 4.2] and, thus, it suffices to take $H = D$.

To prove the "only if" part, assume to the contrary that $V_1 - \ldots - V_n$ is a chordless undirected cycle in $(G_{cl(K)})^m$ for some $K \in cc(G)$. Note that $H$ has the same adjacencies as $G$. Therefore, $V_{i-1} \leftarrow V_i$ and/or $V_i \rightarrow V_{i+1}$ must be in $H$ because, otherwise, $V_{i-1} \perp_G V_{i+1} | Z \in I_{LWF}(G)$ for some $Z$ st $V_i \in Z$ whereas $V_{i-1} \perp_H V_{i+1} | Z \notin I_{MVR}(H)$, which contradicts that $I_{LWF}(G) = I_{MVR}(H)$. Assume without loss of generality that $V_i \rightarrow V_{i+1}$ is in $H$. Then, $V_{i+1} \rightarrow V_{i+2}$ must be in $H$ too, by an argument similar to the previous one. Repeated application of this reasoning implies that $H$ has a semi-directed cycle, which contradicts the definition of CG. □

## Acknowledgments

This work is funded by the Center for Industrial Information Technology (CENIIT) and a so-called career contract at Linköping University, by the Swedish Research Council (ref. 2010-4808), and by FEDER funds and the Spanish Government (MICINN) through the project TIN2010-20900-C04-03.

## References

[1] S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of Markov Equivalence Classes for Acyclic Digraphs. *The Annals of Statistics*, 25:505–541, 1997.

[2] S. A. Andersson, D. Madigan, and M. D. Perlman. An Alternative Markov property for Chain Graphs. *Scandianavian Journal of Statistics*, 28:33–85, 2001.

[3] D. R. Cox and N. Wermuth. Linear Dependencies Represented by Chain Graphs. *Statistical Science*, 8:204–283, 1993.

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

[5] M. Drton. Discrete Chain Graph Models. *Bernoulli*, 15:736–753, 2009.

[6] M. Frydenberg. The Chain Graph Markov Property. *Scandinavian Journal of Statistics*, 17:333–353, 1990.

[7] S. L. Lauritzen and N. Wermuth. Graphical Models for Association Between Variables, Some of Which are Qualitative and Some Quantitative. *The Annals of Statistics*, 17:31–57, 1989.

[8] A. Roverato and M. Studený. A Graphical representation of Equivalence Classes of AMP Chain Graphs. *Journal of Machine Learning Research*, 7:1045–1078, 2006.

[9] D. Sonntag and J. M. Peña. Learning Multivariate Regression Chain Graphs under Faithfulness. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 299–306, 2012.

[10] M. Studený, A. Roverato, and Š. Štěpánová. Two Operations of Merging and Splitting Components in a Chain Graph. *Kybernetika*, 45:208–248, 1997.

[11] N. Wermuth and K. Sadeghi. Sequences of Regression and Their Independences. arXiv:1103.2523 [stat.ME], 2012.

[12] N. Wermuth, M. Wiedenbeck, and D. R. Cox. Partial Inversion for Linear Systems and Partial Closure of Independence Graphs. *BIT Numerical Mathematics*, 46:883–901, 2006.

## 5.4 Article 3: Approximate counting of graphical models via MCMC revisited

### Bibliography:

Title: Approximate Counting of Graphical Models Via MCMC Revisited
Authors: Dag Sonntag, Jose M. Peña and Manuel Gómez-Olmedo
Publication: International Journal of Intelligent Systems (2014)
Status: Under review

# Approximate Counting of Graphical Models Via MCMC Revisited

Dag Sonntag [a], Jose M. Peña [a] and Manuel Gómez-Olmedo [b]

[a] ADIT, IDA, Linköping University, Sweden
[b] Dept. Computer Science and Artificial Intelligence, University of Granada, Spain

**Abstract** We apply MCMC sampling to approximately calculate some quantities, and discuss their implications for learning directed and acyclic graphs (DAGs) from data. Specifically, we calculate the approximate ratio of essential graphs (EGs) to DAGs for up to 31 nodes. Our ratios suggest that the average Markov equivalence class is small. We show that a large majority of the classes seem to have a size that is close to the average size. This suggests that one should not expect more than a moderate gain in efficiency when searching the space of EGs instead of the space of DAGs. We also calculate the approximate ratio of connected EGs to connected DAGs, of connected EGs to EGs, and of connected DAGs to DAGs. These new ratios are interesting because, as we will see, they suggest that some conjectures that appear in the literature do not hold. Furthermore, we prove that the latter ratio is asymptotically 1.

Finally, we calculate the approximate ratio of EGs to largest chain graphs for up to 25 nodes. Our ratios suggest that Lauritzen-Wermuth-Frydenberg chain graphs are considerably more expressive than DAGs. We also report similar approximate ratios and conclusions for multivariate regression chain graphs.
**Keywords:** MCMC sampling, Bayesian networks, Chain graphs, Lauritzen-Wermuth-Frydenberg interpretation, Multivariate regression interpretation

## 1 Introduction

It is well-known that different directed and acyclic graphs (DAGs),[1] also known as Bayesian networks, can represent the same independence model. All such DAGs are said to form a Markov equivalence class. Probably the most common approach to learning DAGs from data is that of performing a search in the space of either DAGs or Markov equivalence classes. In the latter case, the classes are typically represented as essential graphs (EGs).[2]

[1] All the graphs considered in this paper are labeled graphs.

[2] The EG corresponding to a class is the graph that has the directed edge $A \rightarrow B$ iff $A \rightarrow B$ is in every DAG in the class, and the undirected edge $A - B$ iff $A \rightarrow B$ is in some DAGs in the class and $A \leftarrow B$ is in some others [4].

Knowing the ratio of EGs to DAGs for a given number of nodes is a valuable piece of information when deciding which space to search. For instance, if the ratio is low, then one may prefer to search the space of EGs rather than the space of DAGs, though the latter is usually considered easier to traverse. Unfortunately, while the number of DAGs can be computed without enumerating them all [20, Equation 8], the only method for counting EGs that we are aware of is enumeration. Specifically, Gillispie and Perlman enumerated all the EGs for up to 10 nodes by means of a computer program [10]. They showed that the ratio is around 0.27 for 7-10 nodes. They also conjectured a similar ratio for more than 10 nodes by extrapolating the exact ratios for up to 10 nodes.

Enumerating EGs for more than 10 nodes seems challenging: To enumerate all the EGs over 10 nodes, the computer program in [10] needed 2253 hours in a "mid-1990s-era, midrange minicomputer". We obviously prefer to know the exact ratio of EGs to DAGs for a given number of nodes rather than an approximation to it. However, an approximate ratio may be easier to obtain and serve as well as the exact one to decide which space to search. In [17], a Markov chain Monte Carlo (MCMC) approach was proposed to approximately calculate the ratio while avoiding enumerating EGs. The approach consisted of the following steps. First, the author constructed a Markov chain (MC) whose stationary distribution was uniform over the space of EGs for the given number of nodes. Then, the author sampled that stationary distribution and computed the fraction $R$ of EGs containing only directed edges (EDAGs) in the sample. Finally, the author transformed this fraction into the desired approximate ratio of EGs to DAGs as follows: Since $\frac{\#EGs}{\#DAGs}$ can be expressed as $\frac{\#EDAGs}{\#DAGs}\frac{\#EGs}{\#EDAGs}$,[3] then we can approximate it by $\frac{\#EDAGs}{\#DAGs}\frac{1}{R}$ where $\#DAGs$ and $\#EDAGs$ can be computed via [20, Equation 8] and [23, p. 270], respectively. The author reported the so-obtained approximate ratio for up to 20 nodes. The approximate ratios agreed well with the exact ones available in the literature and suggested that the exact ratios are not very low (the approximate ratios were 0.26-0.27 for 7-20 nodes). This suggests that one should not expect more than a moderate gain in efficiency when searching the space of EGs instead of the space of DAGs. Of course, this is a bit of a bold claim since the gain is dictated by the average ratio over the EGs visited during the search and not by the average ratio over all the EGs in the search space. For instance, the gain is not the same if we visit the empty EG, whose ratio is 1, or the complete EG, whose ratio is $1/n!$ for $n$ nodes.[4] Unfortunately, it is impossible to know beforehand which EGs will be visited during the search. Therefore, the best we can do is to draw (bold) conclusions based on the average ratio over all

[3] We use the symbol # followed by a class of graphs to denote the cardinality of the class.

[4] In the latter case, note that there are $n!$ orderings of the nodes in the EG and, thus, there are $n!$ orientations of all the undirected edges in the EG, one per ordering, and none of them has directed cycles.

the EGs in the search space.

In this paper, we extend the work in [17] in three directions. First, we report the approximate ratio of EGs to DAGs for up to 31 nodes. Our ratios are always greater than 0.26, which suggests that the average Markov equivalence class is small. Second, we show that a large majority of the classes seem to have a size that is close to the average size. Third, we report some new approximate ratios. Specifically, we report the approximate ratio of connected EGs to connected DAGs, of connected EGs to EGs, and of connected DAGs to DAGs. These new ratios are interesting because, as we will see, they suggest that some conjectures that appear in the literature do not hold.

In [17], the following question is also addressed. Chain graphs (CGs) are graphs with possibly directed and undirected edges, and no semidirected cycle. Then, CGs extend DAGs and, thus, they can represent at least as many independence models as DAGs. However, unlike DAGs whose interpretation is unique, there are three interpretations of CGs as independence models: The Lauritzen-Wermuth-Frydenberg (LWF) interpretation [15], the multivariate regression (MVR) interpretation [7], and the Andersson-Madigan-Perlman (AMP) interpretation [3]. It should be mentioned that no interpretation subsumes any other, i.e. any interpretation can represent independence models that cannot be represented by the other two interpretations [22]. For any of the three interpretations, knowing the ratio of independence models that can be represented by DAGs to independence models that can be represented by CGs is a valuable piece of information when deciding which class of graphical models to use. For instance, if the ratio is low, then one may prefer to use CGs rather than DAGs, though the latter are easier to manipulate and reason with. Unfortunately, the only method for computing the fraction that we are aware of is by enumerating all the independence models that can be represented by CGs. This is what Volf and Studený did for LWF CGs by means of a computer program [24]. Specifically, it is well-known that different LWF CGs can represent the same independence model. All such CGs are said to form a Markov equivalence class, which is typically represented by the largest CG (LCG) in the class.[5] The computer program in [24] enumerated all the LCGs for up to 5 nodes, which enabled the authors to show the ratio of EGs to LCGs is 1 for 2-3 nodes, 0.93 for 4 nodes, and 0.76 for 5 nodes.

That Volf and Studený ran their computer program only up to 5 nodes indicates that enumerating LCGs is challenging. In [17], a MCMC approach was proposed to approximately calculate the ratio of EGs to LCGs without having to enumerate LCGs. The approach consisted of the following steps. First, the author constructed a MC whose stationary distribution is uniform over the space of LCGs for the given number of nodes. Then, the author sampled this stationary distribution and computed the fraction of the inde-

[5]The LCG in a class is the CG that has the directed edge $A \to B$ iff $A \to B$ is in every CG in the class [8, Proposition 5.7].

pendence models represented by the LCGs in the sample that could also be represented by a DAG. The author reported the so-obtained approximate fraction for up to 13 nodes. The approximate fractions agreed well with the exact ones available in the literature and suggested that the ratio of EGs to LCGs is considerably low (the approximate ratio was 0.04 for 13 nodes). This suggests that one should use CGs instead of DAGs because they are considerably more expressive.

In this paper, we extend the work in [17] in two directions. First, we report the approximate ratio of EGs to LCGs for up to 25 nodes. Second, we report similar approximate ratios for MVR CGs. Our results suggest that both LWF CGs and MVR CGs are considerably more expressive than DAGs.

The rest of the paper is organized as follows. Section 2 presents our results for DAGs. Section 3 presents our results for LWF CGs and MVR CGs. Finally, Section 4 recalls our findings and discusses future work. The paper ends with three appendices devoted to technical details.

# 2 Directed and Acyclic Graphs

## 2.1 Average Markov Equivalence Class Size

In this section, we report the approximate average Markov equivalence class size for 2-31 nodes. To be exact, we report the approximate ratio of EGs to DAGs for 2-31 nodes and, thus, the average class size corresponds to the inverse of the ratio. To obtain the ratios, we run the same computer program implementing the MCMC approach described above as in [17]. The program is written in C++ and compiled in Microsoft Visual C++ 2010 Express. The program is run on an AMD Athlon 64 X2 Dual Core Processor 5000+ 2.6 GHz, 4 GB RAM and Windows Vista Business. The compiler and the computer used in [17] were Microsoft Visual C++ 2008 Express and a Pentium 2.4 GHz, 512 MB RAM and Windows 2000. The experimental settings is the same as in [17] for up to 30 nodes, i.e. each approximate ratio reported is based on a sample of 1e+4 EGs, each obtained as the state of the MC after performing 1e+6 transitions with the empty EG as initial state. For 31 nodes though, each EG sampled is obtained as the state of the MC after performing 2e+6 transitions with the empty EG as initial state. We elaborate later on why we double the length of the MCs for 31 nodes.

Table 1 presents our new approximate ratios $\frac{\#EGs}{\#DAGs}$ and $\frac{\#EDAGs}{\#EGs}$, together with the old approximate ones and the exact ones available in the literature. The first conclusion that we draw from the table is that the new ratios are very close to the exact ones, as well as to the old ones. This makes us confident on the accuracy of the ratios for 11-31 nodes, where no exact ratios are available in the literature due to the high computational cost involved in calculating them. Another conclusion that we draw from the table is that the ratios seem to be 0.26-0.28 for 11-31 nodes. This agrees well

Table 1: Exact and approximate $\frac{\#EGs}{\#DAGs}$ and $\frac{\#EDAGs}{\#EGs}$.

| NODES | EXACT | | | OLD APPROXIMATE | | | NEW APPROXIMATE | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\frac{\#EGs}{\#DAGs}$ | $\frac{\#EDAGs}{\#EGs}$ | Hours | $\frac{\#EGs}{\#DAGs}$ | $\frac{\#EDAGs}{\#EGs}$ | Hours | $\frac{\#EGs}{\#DAGs}$ | $\frac{\#EDAGs}{\#EGs}$ | Hours |
| 2 | 0.66667 | 0.50000 | 0.0 | 0.66007 | 0.50500 | 3.5 | 0.67654 | 0.49270 | 1.3 |
| 3 | 0.44000 | 0.36364 | 0.0 | 0.43704 | 0.36610 | 5.2 | 0.44705 | 0.35790 | 1.0 |
| 4 | 0.34070 | 0.31892 | 0.0 | 0.33913 | 0.32040 | 6.8 | 0.33671 | 0.32270 | 1.2 |
| 5 | 0.29992 | 0.29788 | 0.0 | 0.30132 | 0.29650 | 8.0 | 0.29544 | 0.30240 | 1.4 |
| 6 | 0.28238 | 0.28667 | 0.0 | 0.28118 | 0.28790 | 9.4 | 0.28206 | 0.28700 | 1.6 |
| 7 | 0.27443 | 0.28068 | 0.0 | 0.27228 | 0.28290 | 12.4 | 0.27777 | 0.27730 | 2.0 |
| 8 | 0.27068 | 0.27754 | 0.0 | 0.26984 | 0.27840 | 13.8 | 0.26677 | 0.28160 | 2.3 |
| 9 | 0.26888 | 0.27590 | 7.0 | 0.27124 | 0.27350 | 16.5 | 0.27124 | 0.27350 | 2.6 |
| 10 | 0.26799 | 0.27507 | 2253.0 | 0.26690 | 0.27620 | 18.8 | 0.26412 | 0.27910 | 3.1 |
| 11 | | | | 0.26179 | 0.28070 | 20.4 | 0.26179 | 0.28070 | 3.8 |
| 12 | | | | 0.26737 | 0.27440 | 21.9 | 0.26825 | 0.27350 | 4.2 |
| 13 | | | | 0.26098 | 0.28090 | 23.3 | 0.27405 | 0.26750 | 4.5 |
| 14 | | | | 0.26560 | 0.27590 | 25.3 | 0.27161 | 0.26980 | 5.1 |
| 15 | | | | 0.27125 | 0.27010 | 25.6 | 0.26250 | 0.27910 | 5.7 |
| 16 | | | | 0.25777 | 0.28420 | 27.3 | 0.26943 | 0.27190 | 6.7 |
| 17 | | | | 0.26667 | 0.27470 | 29.9 | 0.26942 | 0.27190 | 7.6 |
| 18 | | | | 0.25893 | 0.28290 | 37.4 | 0.27040 | 0.27090 | 8.2 |
| 19 | | | | 0.26901 | 0.27230 | 38.1 | 0.27130 | 0.27000 | 9.0 |
| 20 | | | | 0.27120 | 0.27010 | 40.3 | 0.26734 | 0.27400 | 9.9 |
| 21 | | | | | | | 0.26463 | 0.27680 | 17.4 |
| 22 | | | | | | | 0.27652 | 0.26490 | 18.8 |
| 23 | | | | | | | 0.26569 | 0.27570 | 13.3 |
| 24 | | | | | | | 0.27030 | 0.27100 | 14.0 |
| 25 | | | | | | | 0.26637 | 0.27500 | 15.9 |
| 26 | | | | | | | 0.26724 | 0.27410 | 17.0 |
| 27 | | | | | | | 0.26950 | 0.27180 | 18.6 |
| 28 | | | | | | | 0.27383 | 0.26750 | 20.1 |
| 29 | | | | | | | 0.27757 | 0.26390 | 21.1 |
| 30 | | | | | | | 0.28012 | 0.26150 | 21.6 |
| 31 | | | | | | | 0.27424 | 0.26710 | 47.3 |

with the conjectured ratio of 0.27 for more than 10 nodes reported in [10]. A last conclusion that we draw from the table is that the fraction of EGs that represent a unique DAG, i.e. $\frac{\#EDAGs}{\#EGs}$, is 0.26-0.28 for 11-31 nodes, a substantial fraction.

Recall from the previous section that we slightly modified the experimental setting for 31 nodes, namely we doubled the length of the MCs. The reason is as follows. We observed an increasing trend in $\frac{\#EGs}{\#DAGs}$ for 25-30 nodes, and interpreted this as an indication that we might be reaching the limits of our experimental setting. Therefore, we decided to double the length of the MCs for 31 nodes in order to see whether this broke the trend. As can be seen in Table 1, it did. This suggests that approximating the ratio for more than 31 nodes will require larger MCs and/or samples than the ones used in this work.

Note that we can approximate the number of EGs for up to 31 nodes as $\frac{\#EGs}{\#DAGs}\#DAGs$, where $\frac{\#EGs}{\#DAGs}$ comes from Table 1 and $\#DAGs$ comes from [20, Equation 8]. Alternatively, we can approximate it as $\frac{\#EGs}{\#EDAGs}\#EDAGs$, where $\frac{\#EGs}{\#EDAGs}$ comes from Table 1 and $\#EDAGs$ can be computed by [23, p. 270].

Table 2: Statistics for the lower and upper bounds.

| NODES | STATISTIC | ARROWS | LINES | LOWER BOUND | UPPER BOUND |
|---|---|---|---|---|---|
| 10 | Minimum | 8 | 0 | 1 | 1 |
| | $Q_1$ | 21 | 0 | 1 | 1 |
| | $Q_2$ | 23 | 1 | 2 | 2 |
| | $Q_3$ | 25 | 3 | 2 | 6 |
| | Maximum | 35 | 19 | 120 | 2073600 |
| 15 | Minimum | 35 | 0 | 1 | 1 |
| | $Q_1$ | 51 | 0 | 1 | 1 |
| | $Q_2$ | 55 | 1 | 2 | 2 |
| | $Q_3$ | 58 | 3 | 2 | 8 |
| | Maximum | 74 | 17 | 240 | 1990656 |
| 20 | Minimum | 72 | 0 | 1 | 1 |
| | $Q_1$ | 94 | 0 | 1 | 1 |
| | $Q_2$ | 99 | 1 | 2 | 2 |
| | $Q_3$ | 104 | 3 | 2 | 6 |
| | Maximum | 127 | 22 | 720 | 2073600 |
| 25 | Minimum | 117 | 0 | 1 | 1 |
| | $Q_1$ | 150 | 0 | 1 | 1 |
| | $Q_2$ | 155 | 1 | 2 | 2 |
| | $Q_3$ | 161 | 3 | 2 | 6 |
| | Maximum | 187 | 17 | 120 | 345600 |
| 30 | Minimum | 90 | 0 | 1 | 1 |
| | $Q_1$ | 218 | 0 | 1 | 1 |
| | $Q_2$ | 224 | 1 | 2 | 2 |
| | $Q_3$ | 231 | 3 | 2 | 6 |
| | Maximum | 265 | 117 | 39916800 | 1.996e+50 |
| 32 | Minimum | 58 | 0 | 1 | 1 |
| | $Q_1$ | 248 | 0 | 1 | 1 |
| | $Q_2$ | 255 | 1 | 2 | 2 |
| | $Q_3$ | 263 | 3 | 2 | 8 |
| | Maximum | 304 | 185 | 8.718e+10 | 8.827e+83 |

Finally, a few words on the running times reported in Table 1 may be in place. First, note that the times reported in Table 1 for the exact ratios are borrowed from [10] and, thus, they correspond to a computer program run on a "mid-1990s-era, midrange minicomputer". Therefore, a direct comparison to our times seems unadvisable. Second, our times are around four times faster than the old times. The reason may be in the use of a more powerful computer and/or a different version of the compiler. The reason cannot be in the difference in the computer programs run, since this is negligible. Third, the new times have some oddities, e.g. the time for two nodes is greater than the time for three nodes. The reason may be that the computer ran other programs while running the experiments reported in this paper.

## 2.2 Variability of the Markov Equivalence Class Size

In the previous section, we have shown that $\frac{\#EDAGs}{\#EGs}$ is approximately 0.26-0.28 for 6-31 nodes. This means that the Markov equivalence class size is approximately 3.6-3.9 on average for 6-31 nodes. In this section, we report on the variability of the class size for 10-32 nodes. Recall that, for $n$ nodes, the class size can vary between 1 and $n!$. However, it has been shown in [10] by enumerating all the DAGs in all the classes for up to 10 nodes, that a

large majority of the classes for a given number of nodes have size $\leq 4$. In this section, we provide evidence that that result may hold for 10-32 nodes too. Therefore, for any number of nodes in the range 2-32, a large majority of the classes seem to have a size relatively close to the average size of 3.6-3.9 and, thus, this average value may be a reasonable estimate of the size of a randomly chosen class.

To arrive at the conclusion above, we ran the same computer program as in the previous section to sample 2e+4 EGs, each obtained as the state of the MC after performing 2e+6 transitions with the empty EG as initial state. The only reason why we doubled parameters as compared to the previous section is because time permitted it. However, time did not permit to compute the sizes of all the classes represented by all the EGs sampled by enumerating all the DAGs in the classes. Therefore, we decided to bound the class sizes instead. A lower bound can be obtained by first finding the largest clique in each connectivity component of the EG and, then, taking the product of the factorials of the sizes of these cliques. An upper bound can be obtained as the product of the factorials of all the cliques in all the connectivity components in the EG. To see how we arrived at these bounds, note that all the connectivity components of an EG are chordal [4, Theorem 4.1]. Then, we can use the algorithm in [14, Theorem 4.13] to orient the undirected edges in each connectivity component st neither immoralities nor directed cycles appear in any former connectivity component. This together with the fact proven in [4, Theorem 4.1] that an EG is a CG that has no induced subgraph of the form $A \rightarrow B - C$ ensure that neither new immoralities nor directed cycles appear in the resulting directed graph. Then, the resulting directed graph is a DAG that belongs to the class represented by the EG. Specifically, the algorithm arranges all the cliques in a connectivity component in a tree, chooses any of them as the root of the tree, and then orients all the undirected edges in any way that respects the following constraint: $A - B$ cannot get oriented as $A \rightarrow B$ if $B$ belongs to a clique that is closer to the root than the clique $A$ belongs to. Therefore, if we choose the largest clique as the root of the clique tree, then we arrive at our lower bound. On the other hand, if we disregard the constraint mentioned when orienting the undirected edges, then we arrive at our upper bound. Our bounds are probably rather loose but, on the other hand, they are easy to compute and, as we will see, tight enough for our purpose.

For each of the 2e+4 EGs sampled, we computed the following statistics: Minimum, maximum, and first, second and third quartiles ($Q_1$, $Q_2$ and $Q_3$) of the number of directed edges (arrows), undirected edges (lines), and lower and upper bounds. The results for 10, 15, 20, 25, 30 and 32 nodes can be seen Table 2. The results for the rest of the numbers of nodes considered are very similar to the ones in the table and, thus, we decided to omit them. The first conclusion that we can draw from the table is that whereas $Q_1$, $Q_2$ and $Q_3$ for the number of arrows grow with the number of nodes, $Q_1$, $Q_2$ and $Q_3$ for the number of lines remain low and constant as the number of

nodes grows. This implies that, as we can see in the table, $Q_1$, $Q_2$ and $Q_3$ for the lower and upper bounds do not vary substantially with the number of nodes. Recall that all the DAGs in a class only differ in the orientation of some of the lines in the corresponding EG. So, if the EG has few lines, then the class must be small. Note however that the maxima of the lower and upper bounds do vary substantially with the number of nodes, specially between 25 and 30 nodes. We interpret this, again, as an indication that 30 nodes might be the limit of our experimental setting. The second conclusion that we can draw is that $Q_1$ for the lower and upper bounds is 1. Therefore, $\geq 25\%$ of the classes sampled have size 1. This is not a surprise because, as shown in the previous section, $\frac{\#EDAGs}{\#EGs}$ which represents the fraction of classes of size 1 is 0.26-0.28 for 7-31 nodes. The third conclusion that we can draw is that $Q_2$ for the lower and upper bounds is 2. Therefore, $\geq 50\%$ of the classes sampled have size $\leq 2$ and, thus, they are smaller than the average size of 3.6-3.9. The fourth conclusion that we can draw is that $Q_3$ for the upper bound is 8. Therefore, $\geq 75\%$ of the classes sampled have size $\leq 8$ and, thus, relatively close to the average size of 3.6-3.9. It is worth mentioning that the last three conclusions agree well with the class size distributions reported in [10] for up to 10 nodes. In summary, despite the class size can vary between 1 and $n!$ for $n$ nodes, our results suggest that a large majority of the classes have a size relatively close to the average size of 3.6-3.9 and, thus, this average value may be a reasonable estimate of the size of a randomly chosen class.

## 2.3 Disproof of Some Conjectures

In this section, we report the approximate ratio of connected EGs (CEGs) to connected DAGs (CDAGs), of CEGs to EGs, and of CDAGs to DAGs. These ratios are interesting because, as we will see, they suggest that some conjectures that appear in the literature do not hold. The approximate ratio of CEGs to CDAGs is computed from the sample obtained in Section 2.1 as follows. First, we compute the ratio $R'$ of EDAGs to CEGs in the sample. Second, we transform this approximate ratio into the desired approximate ratio of CEGs to CDAGs as follows: Since $\frac{\#CEGs}{\#CDAGs}$ can be expressed as $\frac{\#EDAGs}{\#CDAGs}\frac{\#CEGs}{\#EDAGs}$, then we can approximate it by $\frac{\#EDAGs}{\#CDAGs}\frac{1}{R'}$ where $\#EDAGs$ can be computed by [23, p. 270] and $\#CDAGs$ can be computed as shown in Appendix A. The approximate ratio of CEGs to EGs is computed directly from the sample. The approximate ratio of CDAGs to DAGs is computed with the help of Appendix A and [20, Equation 8].

In [10, p. 153], it is stated that "the variables chosen for inclusion in a multivariate data set are not chosen at random but rather because they occur in a common real-world context, and hence are likely to be correlated to some degree". This implies that the EG learnt from some given data is likely to be connected. We agree with this observation, because we believe that humans are good at detecting sets of mutually uncorrelated variables so

Table 3: Approximate $\frac{\#CEGs}{\#CDAGs}$, $\frac{\#CEGs}{\#EGs}$ and $\frac{\#CDAGs}{\#DAGs}$.

| NODES | NEW APPROXIMATE | | |
|---|---|---|---|
| | $\frac{\#CEGs}{\#CDAGs}$ | $\frac{\#CEGs}{\#EGs}$ | $\frac{\#CDAGs}{\#DAGs}$ |
| 2 | 0.51482 | 0.50730 | 0.66667 |
| 3 | 0.39334 | 0.63350 | 0.72000 |
| 4 | 0.32295 | 0.78780 | 0.82136 |
| 5 | 0.29471 | 0.90040 | 0.90263 |
| 6 | 0.28033 | 0.94530 | 0.95115 |
| 7 | 0.27799 | 0.97680 | 0.97605 |
| 8 | 0.26688 | 0.98860 | 0.98821 |
| 9 | 0.27164 | 0.99560 | 0.99415 |
| 10 | 0.26413 | 0.99710 | 0.99708 |
| 11 | 0.26170 | 0.99820 | 0.99854 |
| 12 | 0.26829 | 0.99940 | 0.99927 |
| 13 | 0.27407 | 0.99970 | 0.99964 |
| 14 | 0.27163 | 0.99990 | 0.99982 |
| 15 | 0.26253 | 1.00000 | 0.99991 |
| 16 | 0.26941 | 0.99990 | 0.99995 |
| 17 | 0.26942 | 1.00000 | 0.99998 |
| 18 | 0.27041 | 1.00000 | 0.99999 |
| 19 | 0.27130 | 1.00000 | 0.99999 |
| 20 | 0.26734 | 1.00000 | 1.00000 |
| 21 | 0.26463 | 1.00000 | 1.00000 |
| 22 | 0.27652 | 1.00000 | 1.00000 |
| 23 | 0.26569 | 1.00000 | 1.00000 |
| 24 | 0.27030 | 1.00000 | 1.00000 |
| 25 | 0.26637 | 1.00000 | 1.00000 |
| 26 | 0.26724 | 1.00000 | 1.00000 |
| 27 | 0.26950 | 1.00000 | 1.00000 |
| 28 | 0.27383 | 1.00000 | 1.00000 |
| 29 | 0.27757 | 1.00000 | 1.00000 |
| 30 | 0.28012 | 1.00000 | 1.00000 |
| 31 | 0.27424 | 1.00000 | 1.00000 |
| $\infty$ | ? | ? | $\approx 1$ |

that the original learning problem can be divided into smaller independent learning problems, each of which results in a CEG. Therefore, although we still cannot say which EGs will be visited during the search, we can say that some of them will most likely be connected and some others disconnected. This raises the question of whether $\frac{\#CEGs}{\#CDAGs} \approx \frac{\#DEGs}{\#DDAGs}$ where DEGs and DDAGs stand for disconnected EGs and disconnected DAGs. In [10, p. 154], it is also said that a consequence of the learnt EG being connected is "that a substantial number of undirected edges are likely to be present in the representative essential graph, which in turn makes it likely that the corresponding equivalence class size will be relatively large". In other words, they conjecture that the equivalence classes represented by CEGs are relatively large. We interpret the term "relatively large" as having a ratio smaller than $\frac{\#EGs}{\#DAGs}$. However, this conjecture does not seem to hold according to the approximate ratios presented in Table 3. There, we can see that $\frac{\#CEGs}{\#CDAGs} \approx$ 0.26-0.28 for 6-31 nodes and, thus, $\frac{\#CEGs}{\#CDAGs} \approx \frac{\#EGs}{\#DAGs}$. That the two ratios coincide is not by chance because $\frac{\#CEGs}{\#EGs} \approx$ 0.95-1 for 6-31 nodes, as can be seen in the table. A problem of this ratio being so

close to 1 is that sampling a DEG is so unlikely that we cannot answer the question of whether $\frac{\#CEGs}{\#CDAGs} \approx \frac{\#DEGs}{\#DDAGs}$ with our sampling scheme. Therefore, we have to content with having learnt that $\frac{\#CEGs}{\#CDAGs} \approx \frac{\#EGs}{\#DAGs}$. It is worth mentioning that this result is somehow conjectured by Kočka when he states in a personal communication to Gillispie that "large equivalence classes are merely composed of independent classes of smaller sizes that combine to make a single larger class" [9, p. 1411]. Again, we interpret the term "large" as having a ratio smaller than $\frac{\#EGs}{\#DAGs}$. Again, we cannot check Kočka's conjecture because sampling a DEG is very unlikely. However, we believe that the conjecture holds, because we expect the ratios for those EGs with $k$ connected components to be around $0.27^k$, i.e. we expect the ratios of the components to be almost independent one of another. Gillispie goes on saying that "an equivalence class encountered at any single step of the iterative [learning] process, a step which may involve altering only a small number of edges (typically only one), might be quite small" [9, p. 1411]. Note that the equivalence classes that he suggests that are quite small must correspond to CEGs, because he suggested before that large equivalence classes correspond to DEGs. We interpret the term "quite small" as having a ratio greater than $\frac{\#EGs}{\#DAGs}$. Again, this conjecture does not seem to hold according to the approximate ratios presented in Table 3. There, we can see that $\frac{\#CEGs}{\#CDAGs} \approx$ 0.26-0.28 for 6-31 nodes and, thus, $\frac{\#CEGs}{\#CDAGs} \approx \frac{\#EGs}{\#DAGs}$.

From the results in Tables 1 and 3, it seems that the asymptotic values for $\frac{\#EGs}{\#DAGs}$, $\frac{\#EDAGs}{\#EGs}$, $\frac{\#CEGs}{\#CDAGs}$ and $\frac{\#CEGs}{\#EGs}$ should be around 0.27, 0.27, 0.27 and 1, respectively. It would be nice to have a formal proof of these results. In this paper, we have proven a related result, namely that the ratio of CDAGs to DAGs is asymptotically 1. The proof can be found in Appendix B. Note from Table 3 that the asymptotic value is almost achieved for 6-7 nodes already. Our result adds to the list of similar results in the literature, e.g. the ratio of labeled connected graphs to labeled graphs is asymptotically 1 [12, p. 205].

Note that we can approximate the number of CEGs for up to 31 nodes as $\frac{\#CEGs}{\#EGs}\#EGs$, where $\frac{\#CEGs}{\#EGs}$ comes from Table 3 and $\#EGs$ can be computed as shown in the previous section. Alternatively, we can approximate it as $\frac{\#CEGs}{\#CDAGs}\#CDAGs$, where $\frac{\#CEGs}{\#CDAGs}$ comes from Table 3 and $\#CDAGs$ can be computed as shown in Appendix A.

# 3 Chain Graphs

Chain graphs (CGs) is a class of graphical models containing two types of edges, directed edges and a secondary type of edge. The secondary type of edge is then used to create components in the graph that are connected by directed edges similarly as the nodes in a DAG. This allows CGs to represents a much larger set of independence models compared to DAGs, while still keeping some of the simplicity that makes DAGs so useful.

As noted in the introduction, there exist three interpretations of how to read independencies from a CG. However, the three coincide when the CG has only directed edges. Hence, DAGs are a subclass of the three CG interpretations. It should also be noted that CGs of the LWF interpretation (LWF CGs) and AMP interpretations (AMP CGs) are typically represented with directed and undirected edges, while CGs of the MVR interpretation (MVR CGs) are typically represented with directed and bidirected edges. LWF CGs and AMP CGs containing only undirected edges are also called Markov networks (MNs), whereas MVR CGs containing only bidirected edges are also called covariance graphs (covGs).

In this work, we have chosen to focus on the LWF and MVR interpretations of CGs. More specifically, we study the ratio of independence models that can be represented by MNs and DAGs (resp. covGs and DAGs) to the independence models that can be represented by LWF CGs (resp. MVR CGs). Hereinafter, these ratios are denoted as $R_{LWFtoMNs}$, $R_{LWFtoDAGs}$, $R_{MVRtoCovGs}$ and $R_{MVRtoDAGs}$. Knowing these ratios is a valuable piece of information when deciding which class of graphical models to use. If a ratio is large (close to 1) then the gain of using the more complex CGs is small compared to using the simpler subclass, while if it is small then one might prefer to using CGs instead of the simpler subclass.

As mentioned in the introduction, the ratios described above have previously been approximated for LWF CGs for up to 13 nodes [17]. Specifically, the author used a MCMC sampling scheme to sample the space of largest chain graphs (LCGs), because each LCG represents one and only one of the independence models that can be represented by LWF CGs. To our knowledge, we are the first to propose a similar scheme to sample the space of independence models that can be represented by MVR CGs. To do so, we had to overcome some major problems. First of all, there existed no unique representative for each independence model that can be represented by MVR CGs. Hence, one such representative, called essential MVR CGs, had to be defined and characterized. Secondly, no operations for creating a Markov chain (MC) over essential MVR CGs existed. This meant that such a set of operations had to be defined so that it could be proven that the stationary distribution of the MC was the uniform distribution. Hence the major contributions in this section, apart from the calculated ratios and their implications, are the essential MVR CGs and the corresponding MC operators.

The rest of the section is structured as follows. First, we define essential MVR CGs and the corresponding MC operators. Then, we move on to the results with a short discussion. Finally, we also discuss some open questions. We have chosen to move theoretical proofs and some additional material for essential MVR CGs to Appendix C.

## 3.1 Essential MVR CGs

As with DAGs and LWF CGs, different MVR CGs can represent the same independence model. All such MVR CGs are said to form a Markov equivalence class. As we have discussed, EGs were introduced to represent Markov equivalence classes of DAGs. EGs were then extended by Ali et al. to represent Markov equivalence classes of so-called ancestral graphs [2]. Since ancestral graphs are a superset of MVR CGs, we know on the one hand that all results presented by Ali et al. also holds for MVR CGs, and hence the extended EGs can be used to represent Markov equivalence classes of MVR CGs. On the other hand, we also know that more restrictions and characteristics might be (and is) possible to assert on the structure of the extended EGs if we only consider MVR CGs rather than ancestral graphs. That is why we define an essential MVR CG as follows.

**Definition 1.** *A graph $G^*$ is said to be the essential MVR CG of a MVR CG $G$ if it has the same skeleton as $G$ and contains all and only the arrowheads common to every MVR CG in the Markov equivalence class of $G$.*

From this definition it is clear that an essential MVR CG is an unique representation of a Markov equivalence class of MVR CGs. Note that an essential MVR CG is not actually a MVR CG but might contain undirected edges as well. Each undirected edge implies that there exists two MVR CGs in the Markov equivalence class that have a directed edge between the same two nodes but in opposite directions. Hence, essential MVR CGs contain components containing only undirected edges (undirected components) as well as components containing only bidirected edges (bidirected components). It can be also be shown that no node can be an endnode of both an undirected edge and a bidirected edge and hence these components are connected to each other with directed edges similarly as the components of any other CG.

Using the separation criterion defined in Appendix C, we can state the following theorem.

**Theorem 1.** *An essential MVR CG $G^*$ represents the same independence model as every MVR CG $G$ it is the essential for.*

Finally, if we define an indifferent arrowhead on an edge as that any graph not containing the arrowhead on that edge would represent another independence model or not fulfill the other criteria for being an essential MVR CG, then we can give a characterization of essential MVR CGs.

**Theorem 2.** *A graph $G$ containing bidirected edges, directed edges and/or undirected edges is an essential MVR CG iff (1) it contains no semi-directed cycles, (2) all arrowheads are indifferent, (3) all undirected components are chordal, and (4) all nodes in the same undirected component share the same parents but have no spouses.*

Finally, we can state the following lemmas.

**Lemma 1.** *There exists a DAG representing the same independence model as an essential MVR CG $G^*$ iff $G^*$ contains no bidirected edges. Moreover, the essential MVR CG is then an EG.*

**Lemma 2.** *There exists a covG representing the same independence model as an essential MVR CG $G^*$ iff every non-collider in $G^*$ is shielded.*

The definitions of the terms and the proofs of these theorems as well as an algorithm for finding the essential MVR CG of a certain MVR CG can be found in Appendix C.

## 3.2 MC operations

In this section, we propose eight operators that can be used to create a MC whose stationary distribution is the uniform distribution over the space of essential MVR CGs for a given number of nodes. Let $G_i$ be the essential MVR CG before the operation and $G_{i+1}$ the essential MVR CG after the operation. The operators are the following ones.

**Definition 2.** *MC Operators*

1. *Add Undirected Edge Choose two nodes $X$ and $Y$ in $G_i$ uniformly and with replacement. If the two nodes are non-adjacent in $G_i$ and the graph resulting from adding an undirected edge between them is an essential MVR CG, then let $G_{i+1} = G_i \cup \{X - Y\}$, otherwise let $G_{i+1} = G_i$.*
2. *Remove Undirected Edge Choose two nodes $X$ and $Y$ in $G_i$ uniformly and with replacement. If the two nodes are neighbours in $G_i$ and the graph resulting from removing the undirected edge between them is an essential MVR CG, then let $G_{i+1} = G_i \setminus \{X - Y\}$, otherwise let $G_{i+1} = G_i$.*
3. *Add Directed Edge Choose two nodes $X$ and $Y$ in $G_i$ uniformly and with replacement. If the two nodes are non-adjacent in $G_i$ and the graph resulting from adding a directed edge from $X$ to $Y$ is an essential MVR CG, then let $G_{i+1} = G_i \cup \{X \to Y\}$, otherwise let $G_{i+1} = G_i$.*
4. *Remove Directed Edge Choose two nodes $X$ and $Y$ in $G_i$ uniformly and with replacement. If there exist a directed edge $X \to Y$ between them in $G_i$ and the graph resulting from removing the directed edge between them is an essential MVR CG, then let $G_{i+1} = G_i \setminus \{X \to Y\}$, otherwise let $G_{i+1} = G_i$.*
5. *Add Bidirected Edge Choose two nodes $X$ and $Y$ in $G_i$ uniformly and with replacement. If the two nodes are non-adjacent in $G_i$ and the graph resulting from adding a bidirected edge between them is an essential MVR CG, then let $G_{i+1} = G_i \cup \{X \leftrightarrow Y\}$, otherwise let $G_{i+1} = G_i$.*

6. *Remove Bidirected Edge* *Choose two nodes $X$ and $Y$ in $G_i$ uniformly and with replacement. If the two nodes are spouses in $G_i$ and the graph resulting from removing the bidirected edge between them is an essential MVR CG, then let $G_{i+1} = G_i \setminus \{X-Y\}$, otherwise let $G_{i+1} = G_i$.*
7. *Add V-collider* *Chose a node $X$ in $G_i$ uniformly. If $|ad_{G_i}(X)| \geq 2$,[6] let $k = rand(1, |ad_{G_i}(X)|)$ and let $V_k$ be $k$ nodes taken uniformly from $ad_{G_i}(X)$ with replacement. If all edge-endings towards $X$ from every node in $V_k$ are non-arrows, and the graph resulting from replacing these edge-endings with arrows is an essential MVR CG, then let $G_{i+1}$ be such a graph. Otherwise let $G_{i+1} = G_i$.*
8. *Remove V-collider* *Chose a node $X$ in $G_i$ uniformly. If $|ad_{G_i}(X)| \geq 2$, let $k = rand(1, |ad_{G_i}(X)|)$ and let $V_k$ be $k$ nodes taken uniformly from $ad_{G_i}(X)$ with replacement. If all edge-endings towards $X$ from every node in $V_k$ are arrows, and the graph resulting from replacing these edge-endings with non-arrows is an essential MVR CG, then let $G_{i+1}$ be such a graph. Otherwise let $G_{i+1} = G_i$.*

**Theorem 3.** *The MC created from the operators in Definition 2 reaches the uniform distribution over the space of essential MVR CGs for the given number of nodes when the number of transitions goes to infinity.*

## 3.3 Results

Using the MC operations described in [17] and those described above, LCGs and essential MVR CGs were sampled and the above described ratios calculated. Specifically, 1e+5 LCGs and 1e+5 essential MVR CGs were sampled with 1e+5 transitions between each sample. The sampling was performed on a standard desktop computer (Intel core I5 2,8Gh with 4 GB memory) and took approximately 11 hours for 25 nodes. To check if the independence model represented by a LCG could be represented by a DAG or MN, we made use of the results in [5, Propositions 4.1 and 4.2]. To check if the independence model represented by an essential MVR CG could be represented by a DAG or covG, we made use of Lemma 1 and Lemma 2.

The calculated approximate ratios can be found in Tables 4 and 5. In addition to these, we also present the exact ratios found through enumeration for up to 5 nodes. Finally we have also added a third ratio, $R_{LWFpureCGs}$ resp. $R_{MVRpureCGs}$, describing the ratio of pure LCGs resp. pure essential MVR CGs to all independence models representable by the corresponding interpretation. A pure LCG resp. essential MVR CG represents an independence model that cannot be represented by any DAG or MN resp. DAG or covG. Note that this is not equal to all the independence models that can be represented by LWF CGs minus those that can be represented by DAGs or MNs, since some models can be represented by both DAGs and MNs (and similarly for MVR CGs).

[6] With $ad_{G_i}(X)$, we mean the nodes adjacent to $X$ in the graph $G_i$.

Table 4: Exact and approximate $R_{LWFtoDAGs}$, $R_{LWFtoMNs}$ and $R_{LWFpureCGs}$ (in this order).

| NODES | EXACT | | | APPROXIMATE | | |
|---|---|---|---|---|---|---|
| 2 | 1.00000 | 1.00000 | 0.00000 | 1.00000 | 1.00000 | 0.00000 |
| 3 | 0.72727 | 1.00000 | 0.00000 | 0.71883 | 1.00000 | 0.00000 |
| 4 | 0.32000 | 0.92500 | 0.06000 | 0.31217 | 0.93266 | 0.05671 |
| 5 | 0.08890 | 0.76239 | 0.22007 | 0.08093 | 0.76462 | 0.21956 |
| 6 | | | | 0.01650 | 0.58293 | 0.40972 |
| 7 | | | | 0.00321 | 0.41793 | 0.57975 |
| 8 | | | | 0.00028 | 0.28602 | 0.71375 |
| 9 | | | | 0.00018 | 0.19236 | 0.80746 |
| 10 | | | | 0.00001 | 0.12862 | 0.87137 |
| 11 | | | | 0.00000 | 0.08309 | 0.91691 |
| 12 | | | | 0.00000 | 0.05544 | 0.94456 |
| 13 | | | | 0.00000 | 0.03488 | 0.96512 |
| 14 | | | | 0.00000 | 0.02371 | 0.97629 |
| 15 | | | | 0.00000 | 0.01518 | 0.98482 |
| 16 | | | | 0.00000 | 0.00963 | 0.99037 |
| 17 | | | | 0.00000 | 0.00615 | 0.99385 |
| 18 | | | | 0.00000 | 0.00382 | 0.99618 |
| 19 | | | | 0.00000 | 0.00267 | 0.99733 |
| 20 | | | | 0.00000 | 0.00166 | 0.99834 |
| 21 | | | | 0.00000 | 0.00105 | 0.99895 |
| 22 | | | | 0.00000 | 0.00079 | 0.99921 |
| 23 | | | | 0.00000 | 0.00035 | 0.99965 |
| 24 | | | | 0.00000 | 0.00031 | 0.99969 |
| 25 | | | | 0.00000 | 0.00021 | 0.99979 |

Regarding the accuracy of the approximations, we can see that in both tables the approximations agree well with the exact values. Moreover, plotting the approximated values results in smooth curves, indicating that the LCGs and essential MVR CGs were sampled from an almost uniform distribution. This is further supported by plots of the average number of directed edges, undirected edges or bidirected edges. We omit this results for brevity. For more than 25 nodes, we could however notice inconsistencies in the approximations indicating that not enough MC transitions were performed.

Regarding the approximate ratios themselves, we can see that $R_{LWFtoDAGs}$ and $R_{MVRtoDAGs}$ decrease exponentially when the number of nodes grows. This agrees well with previous results. However, we are the first to identify this trend. Specifically, for more than three nodes, the approximate (and exact) $R_{LWFtoDAGs}$ resp. $R_{MVRtoDAGs}$ almost perfectly follows the curve $9.379 * 0.6512^n$ resp. $5.352 * 0.6614^n$ where $n$ is the number of nodes. Moreover, as $R_{LWFtoMNs}$ and $R_{MVRtoCovGs}$ suggest, MNs and covGs can only represent a very small set of the independence models that CGs and also DAGs can represent. Already for 10 nodes the ratios are $\leq$ 1e-5 and hence, since only 1e+5 graphs were sampled, they are unreliable. Finally, we can see that $R_{LWFpureCGs}$ and $R_{MVRpureCGs}$ grow very fast so that they are $\geq 0.99$ for already 15 nodes. Hence, this indicates that there is a large gain in using the more advanced class of CGs compared to DAGs, MNs or covGs.

Note that we can obtain approximate numbers of LCGs and essential

Table 5: Exact and approximate $R_{MVRtoDAGs}$, $R_{MVRtoCovGs}$ and $R_{MVRpureCGs}$ (in this order).

| NODES | EXACT | | | APPROXIMATE | | |
|---|---|---|---|---|---|---|
| 2 | 1.00000 | 1.00000 | 0.00000 | 1.00000 | 1.00000 | 0.00000 |
| 3 | 0.54545 | 1.00000 | 0.00000 | 0.72547 | 1.00000 | 0.00000 |
| 4 | 0.10714 | 0.82589 | 0.10714 | 0.28550 | 0.82345 | 0.10855 |
| 5 | 0.00807 | 0.59074 | 0.36762 | 0.06967 | 0.59000 | 0.36787 |
| 6 | | | | 0.01241 | 0.40985 | 0.57921 |
| 7 | | | | 0.00187 | 0.28675 | 0.71145 |
| 8 | | | | 0.00028 | 0.19507 | 0.80465 |
| 9 | | | | 0.00002 | 0.13068 | 0.86930 |
| 10 | | | | 0.00000 | 0.08663 | 0.91337 |
| 11 | | | | 0.00000 | 0.05653 | 0.94347 |
| 12 | | | | 0.00000 | 0.03771 | 0.96229 |
| 13 | | | | 0.00000 | 0.02385 | 0.97615 |
| 14 | | | | 0.00000 | 0.01592 | 0.98408 |
| 15 | | | | 0.00000 | 0.00983 | 0.99017 |
| 16 | | | | 0.00000 | 0.00644 | 0.99356 |
| 17 | | | | 0.00000 | 0.00485 | 0.99515 |
| 18 | | | | 0.00000 | 0.00267 | 0.99733 |
| 19 | | | | 0.00000 | 0.00191 | 0.99809 |
| 20 | | | | 0.00000 | 0.00112 | 0.99888 |
| 21 | | | | 0.00000 | 0.00073 | 0.99927 |
| 22 | | | | 0.00000 | 0.00048 | 0.99952 |
| 23 | | | | 0.00000 | 0.00035 | 0.99965 |
| 24 | | | | 0.00000 | 0.00017 | 0.99983 |
| 25 | | | | 0.00000 | 0.00014 | 0.99986 |

MVR CGs for up to 25 nodes by just multiplying the inverse of $R_{LWFtoMNs}$ and $R_{MVRtoCovGs}$ by the corresponding number of independence models that can be represented by MNs and covGs, which is $2^{\frac{n(n-1)}{2}}$ for $n$ nodes. Alternatively, we can multiply the inverse of $R_{LWFtoDAGs}$ and $R_{MVRtoDAGs}$ by the numbers of EGs, which are known exactly for up to 10 nodes or can be estimated for up to 31 nodes as we have described in Section 2.1.

# 4 Discussion

In [10], it is shown that $\frac{\#EGs}{\#DAGs} \approx 0.27$ for 7-10 nodes. We have shown in this paper that $\frac{\#EGs}{\#DAGs} \approx$ 0.26-0.28 for 11-31 nodes. These results indicate that the average Markov equivalence class size is 3.6-3.9 and, thus, one should not expect more than a moderate gain in efficiency when searching the space of EGs instead of the space of DAGs. We have also shown that a large majority of the classes have a size relatively close to the average size of 3.6-3.9 and, thus, this average value may be a reasonable estimate of the size of a randomly chosen class. We have also shown that $\frac{\#CEGs}{\#CDAGs} \approx$ 0.26-0.28 for 6-31 nodes and, thus, $\frac{\#CEGs}{\#CDAGs} \approx \frac{\#EGs}{\#DAGs}$. Therefore, when searching the space of EGs, the fact that some of the EGs visited will most likely be connected does not seem to imply any additional gain in efficiency beyond that due to searching the space of EGs instead of the space of DAGs.

Some questions that remain open and that we would like to address in

the future are checking whether $\frac{\#CEGs}{\#CDAGs} \approx \frac{\#DEGs}{\#DDAGs}$, and computing the asymptotic ratios of EGs to DAGs, EDAGs to EGs, CEGs to CDAGs, and of CEGs to EGs. Recall that in this paper we have proven that the asymptotic ratio of CDAGs to DAG is 1. Another topic for further research, already mentioned in [17], would be improving the graphical modifications that determine the MC transitions, because they rather often produce a graph that is not an EG. Specifically, the MC transitions are determined by choosing uniformly one out of seven modifications to perform on the current EG. Actually, one of the modifications leaves the current EG unchanged. Therefore, around 14 % of the modifications cannot change the current EG and, thus, 86 % of the modifications can change the current EG. In our experiments, however, only 6-8 % of the modifications change the current EG. The rest up to the mentioned 86 % produce a graph that is not an EG and, thus, they leave the current EG unchanged. This problem has been previously pointed out in [18]. Furthermore, he presents a set of more complex modifications that are claimed to alleviate the problem just described. Unfortunately, no evidence supporting this claim is provided. More recently, He et al. have proposed an alternative set of modifications having a series of desirable features that ensure that applying the modifications to an EG results in a different EG [13]. Although these modifications are more complex than those in [17], the authors show that their MCMC approach is thousands of times faster for 3, 4 and 6 nodes [13, pp. 17-18]. However, they also mention that it is unfair to compare these two approaches: Whereas 1e+4 MCs of 1e+6 transitions each are run in [17] to obtain a sample, they only run one MC of between 1e+4 and 1e+5 transitions. Therefore, it is not clear how their MCMC approach scales to 10-30 nodes as compared to the one in [17]. The point of developing modifications that are more effective than ours at producing EGs is to make a better use of the running time by minimizing the number of graphs that have to be discarded. However, this improvement in effectiveness has to be weighed against the computational cost of the modifications, so that the MCMC approach still scales to the number of nodes of interest.

In this paper, we have also studied the LWF and MVR interpretations of CGs and shown that only a very small portion of the independence models represented by these can be represented by DAGs, MNs or covGs. More specifically, we have identified that this ratio decreases exponentially when the number of nodes grows. During the process to obtain this results, we have defined and characterized a unique representative for each independence model representable by MVR CGs similar to LCGs for LWF CGs and EGs for DAGs. This allows for future research in what independence models MVR CGs can represent as well how the number of members varies in the different Markov equivalence classes of MVR CGs. In the future, it would also be interesting to look further on how the results for these two interpretations relate to similar results for the AMP interpretation of CGs. Apart from these topics, a future follow-up work could of course consider

larger number of nodes. For the MC operations described here for CGs, only about 8 % of the MC transitions were successful, similarly as noted for DAGs. In our experiments, we also observed that the majority of the runtime was spent on checking whether the modified graphs were LCGs or essential MVR CGs. Both these checks can however be done in polynomial time but they can of course be improved, e.g. it may be the case that the check can be done locally depending on the operation applied. One problem with increasing the number of nodes in the experiments is however that $R_{LWFtoDAGs}$ and $R_{MVRtoDAGs}$ decrease exponentially with the number of nodes. Hence, to get good approximations, the number of graphs sampled would also have to be increased exponentially. For instance, according to the equations fitted to the approximate ratios reported, we can estimate that $R_{MVRtoDAGs}$ is approximately 2e-5 for 30 nodes, and hence only two essential MVR CGs whose independence models could be represented by a DAGs would be sampled for a sample size of $1e+5$.

# Appendix A: Counting CDAGs

Let $A(x)$ denote the exponential generating function for DAGs. That is,

$$A(x) = \sum_{k=1}^{\infty} \frac{A_k}{k!} x^k$$

where $A_k$ denotes the number of DAGs of order $k$. Likewise, let $a(x)$ denote the exponential generating function for CDAGs. That is,

$$a(x) = \sum_{k=1}^{\infty} \frac{a_k}{k!} x^k$$

where $a_k$ denotes the number of CDAGs of order $k$. Note that $A_k$ can be computed without having to resort to enumeration by [20, Equation 8]. However, we do not know of any formula to compute $a_k$ without enumeration. Luckily, $a_k$ can be computed from $A_k$ as follows. First, note that

$$1 + A(x) = e^{a(x)}$$

as shown by [12, pp. 8-9]. Now, let us define $A_0 = 1$ and redefine $A(x)$ as

$$A(x) = \sum_{k=0}^{\infty} \frac{A_k}{k!} x^k,$$

i.e. the summation starts with $k = 0$. Then,

$$A(x) = e^{a(x)}.$$

Consequently,

$$\frac{a_n}{n!} = \frac{A_n}{n!} - \left(\sum_{k=1}^{n-1} k \frac{a_k}{k!} \frac{A_{n-k}}{(n-k)!}\right)/n$$

as shown by [12, pp. 8-9], and thus

$$a_n = A_n - (\sum_{k=1}^{n-1} k\binom{n}{k} a_k A_{n-k})/n.$$

See also [6, pp. 38-39]. Moreover, according to [1, Sequence A082402], the result in this appendix has previously been reported in [19]. However, we could not gain access to that paper to confirm it.

# Appendix B: Asymptotic Behavior of CDAGs

**Theorem 4.** *The ratio of CDAGs of order $n$ to DAGs of order $n$ tends to 1 as $n$ tends to infinity.*

*Proof.* Let $A_n$ and $a_n$ denote the numbers of DAGs and CDAGs of order $n$, respectively. Specifically, we prove that $(A_n/n!)/(a_n/n!) \to 1$ as $n \to \infty$. By [26, Theorem 6], this holds if the following three conditions are met:

(i) $\log((A_n/n!)/(A_{n-1}/(n-1)!)) \to \infty$ as $n \to \infty$,

(ii) $\log((A_{n+1}/(n+1)!)/(A_n/n!)) \geq \log((A_n/n!)/(A_{n-1}/(n-1)!))$ for all large enough $n$, and

(iii) $\sum_{k=1}^{\infty}(A_k/k!)^2/(A_{2k}/(2k)!)$ converges.

We start by proving that the condition (i) is met. Note that from every DAG $G$ over the nodes $\{v_1, \ldots, v_{n-1}\}$ we can construct $2^{n-1}$ different DAGs $H$ over $\{v_1, \ldots, v_n\}$ as follows: Copy all the arrows from $G$ to $H$ and make $v_n$ a child in $H$ of each of the $2^{n-1}$ subsets of $\{v_1, \ldots, v_{n-1}\}$. Therefore,

$$\log((A_n/n!)/(A_{n-1}/(n-1)!)) \geq \log(2^{n-1}/n)$$

which clearly tends to infinity as $n$ tends to infinity.

We continue by proving that the condition (ii) is met. Every DAG over the nodes $V \cup \{w\}$ can be constructed from a DAG $G$ over $V$ by adding the node $w$ to $G$ and making it a child of a subset $Pa$ of $V$. If a DAG can be so constructed from several DAGs, we simply consider it as constructed from one of them. Let $H_1, \ldots, H_m$ represent all the DAGs so constructed from $G$. Moreover, let $Pa_i$ denote the subset of $V$ used to construct $H_i$ from $G$. From each $Pa_i$, we can now construct $2m$ DAGs over $V \cup \{w, u\}$ as follows: (i) Add the node $u$ to $H_i$ and make it a child of each subset $Pa_j \cup \{w\}$ with $1 \leq j \leq m$, and (ii) add the node $u$ to $H_i$ and make it a parent of each subset $Pa_j \cup \{w\}$ with $1 \leq j \leq m$. Therefore, $A_{n+1}/A_n \geq 2A_n/A_{n-1}$ and thus

$$\log((A_{n+1}/(n+1)!)/(A_n/n!)) = \log(A_{n+1}/A_n) - \log(n+1)$$

$$\geq \log(2A_n/A_{n-1}) - \log(n+1) \geq \log(2A_n/A_{n-1}) - \log(2n) = \log(A_n/A_{n-1}) - \log n$$

$$= \log((A_n/n!)/(A_{n-1}/(n-1)!)).$$

Finally, we prove that the condition (iii) is met. Let $G$ and $G'$ denote two (not necessarily distinct) DAGs of order $k$. Let $V = \{v_1, \ldots, v_k\}$ and $V' = \{v'_1, \ldots, v'_k\}$ denote the nodes in $G$ and $G'$, respectively. Consider the DAG $H$ over $V \cup V'$ that has the union of the arrows in $G$ and $G'$. Let $w$ and $w'$ denote two nodes in $V$ and $V'$, respectively. Let $S$ be a subset of size $k-1$ of $V \cup V' \setminus \{w, w'\}$. Now, make $w$ a parent in $H$ of all the nodes in $S \cap V'$, and make $w'$ a child in $H$ of all the nodes in $S \cap V$. Note that the resulting $H$ is a DAG of order $2k$. Note that there are $k^2$ different pairs of nodes $w$ and $w'$. Note that there are $\binom{2k-2}{k-1}$ different subsets of size $k-1$ of $V \cup V' \setminus \{w, w'\}$. Note that every choice of DAGs $G$ and $G'$, nodes $w$ and $w'$, and subset $S$ gives rise to a different DAG $H$. Therefore, $A_{2k}/A_k^2 \geq k^2\binom{2k-2}{k-1}$ and thus

$$\sum_{k=1}^{\infty} (A_k/k!)^2/(A_{2k}/(2k)!) = \sum_{k=1}^{\infty} A_k^2(2k)!/(A_{2k}k!^2)$$

$$\leq \sum_{k=1}^{\infty} ((k-1)!(k-1)!(2k)!)/(k^2(2k-2)!k!^2) = \sum_{k=1}^{\infty} (4k-2)/k^3$$

which clearly converges. □

# Appendix C: Proof for Section 3

In this appendix we give the proofs for the theorems defined in Section 3. These proofs do however require some more notation to be defined. Hence this section will start with a notation subsection. This is then followed by the proofs for essential MVR CGs and finally the proof of Theorem 3 is given.

## Notation

All graphs are defined over a finite set of variables $V$. If a graph $G$ contains an edge between two nodes $V_1$ and $V_2$, we denote with $V_1 \to V_2$ a *directed edge*, with $V_1 \leftrightarrow V_2$ a *bidirected edge* and with $V_1 - V_2$ an *undirected edge*. By $V_1 \circ\!\!\to V_2$ we mean that either $V_1 \to V_2$ or $V_1 \leftrightarrow V_2$ is in $G$. By $V_1 \circ\!\!- V_2$ we mean that either $V_1 \to V_2$ or $V_1 - V_2$ is in $G$. By $V_1 \circ\!\!-\!\!\circ V_2$ we mean that there exists an edge between $V_1$ and $V_2$ in $G$. A set of nodes is said to be *complete* if there exist edges between all pairs of nodes in the set. Moreover we say that the edge $V_1 \leftarrow V_2$ has an *arrowhead* towards $V_1$ and a *non-arrowhead* towards $V_2$.

The *parents* of a set of nodes $X$ of $G$ is the set $pa_G(X) = \{V_1 | V_1 \to V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *children* of $X$ is the set $ch_G(X) = \{V_1 | V_2 \to V_1$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *spouses* of $X$ is the set $sp_G(X) = \{V_1 | V_1 \leftrightarrow V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *neighbours* of $X$ is the set $nb_G(X) = \{V_1 | V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The *boundary* of $X$ is the set $bd_G(X) = pa_G(X) \cup nb_G(X) \cup sp_G(X)$. The *adjacents* of $X$

is the set $ad_G(X) = \{V_1 | V_1 \to V_2, V_1 \leftarrow V_2, V_1 \leftrightarrow V_2$ or $V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$.

A *route* from a node $V_1$ to a node $V_n$ in $G$ is a sequence of nodes $V_1, \ldots, V_n$ such that $V_i \in ad_G(V_{i+1})$ for all $1 \leq i < n$. A *path* is a route containing only distinct nodes. The length of a path is the number of edges in the path. A path is called a *cycle* if $V_n = V_1$. A path is *descending* if $V_i \in pa_G(V_{i+1}) \cup sp_G(V_{i+1}) \cup nb_G(V_{i+1})$ for all $1 \leq i < n$. A path $\pi = V_1, \ldots, V_n$ is *minimal* if there exists no other path $\pi_2$ between $V_1$ and $V_n$ st $\pi_2 \subset \pi$ holds. The *descendants* of a set of nodes $X$ of $G$ is the set $de_G(X) = \{V_n |$ there is a descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. A path is *strictly descending* if $V_i \in pa_G(V_{i+1})$ for all $1 \leq i < n$. The *strict descendants* of a set of nodes $X$ of $G$ is the set $sde_G(X) = \{V_n |$ there is a strict descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. The *ancestors* (resp. *strict ancestors*) of $X$ is the set $an_G(X) = \{V_1 | V_n \in de_G(V_1), V_1 \notin X, V_n \in X\}$ (resp. $san_G(X) = \{V_1 | V_n \in sde_G(V_1), V_1 \notin X, V_n \in X\}$). A cycle is called a *semi-directed cycle* if it is descending and $V_i \to V_{i+1}$ is in $G$ for some $1 \leq i < n$.

An undirected (resp. bidirected) *component* $C$ of a graph is a maximal (wrt set inclusion) set of nodes such that there exists a path between every pair of nodes in $C$ containing only undirected edges (resp. bidirected edges). If the type (undirected resp. bidirected) is not specified we mean either type of component. We denote the set of all connectivity components in a graph $G$ by $cc(G)$ and the component to which a set of nodes $X$ belong in $G$ by $co_G(X)$. A *subgraph* of $G$ is a subset of nodes and edges in $G$. A subgraph of $G$ induced by a set of its nodes $X$ is the graph over $X$ that has all and only the edges in $G$ whose both ends are in $X$.

In this appendix we deal with three classes of graphs; directed acyclic graphs (DAGs), multivariate regression chain graphs (MVR CGs) and joined chain graphs (JCGs). A DAG contains only directed edges and no semi-directed cycles. A MVR CG is a graph containing only directed and bidirected edges but no semi-directed cycles. JCGs are a subclass of joined graphs that were introduced by Ali et al. in 2005 [2]. Ali et al. define joined graphs to be graphs created by joining ancestral graphs. JCGs have a similar definition with the exception that the graphs joined must be MVR CGs of the same Markov equivalence class. Hence we can define JCGs as:

**Definition 3.** *Joined Chain Graph If $G_1$ and $G_2$ are two MVR CGs belonging to the same Markov equivalence class then define the joined chain graph $G = G_1 \vee G_2$ to be the graph with the same adjacencies such that, on an edge between $X$ and $Y$,*

1. *there is an arrowhead towards $X$ in $G$ only if there is an arrowhead towards $X$ in both $G_1$ and $G_2$.*
2. *there is a no arrowhead towards $X$ in $G$ if there is no arrowhead in towards $X$ in either $G_1$ or $G_2$.*

Let $X$, $Y$ and $Z$ denote three disjoint subsets of $V$. We say that $X$

*separated* (in the classes of graphs described above) from $Y$ given $Z$ denoted as $X \perp_G Y|Z$ iff there exists no d-connecting path between $X$ and $Y$. A path is said to be *d-connecting* iff (1) every non-collider on the path is not in $Z$, (2) every collider on the path is in $Z$ or $san_G(Z)$ and (3) no arrowheads meet any undirected edges. A node $B$ is said to be a *collider* in a JCG, MVR CG or DAG $G$ between two nodes $A$ and $C$ on a path if one of following configuration exists in $G$: $A \circ\!\!\rightarrow B \leftarrow\!\!\circ C$ while for any other configuration ($A \circ\!\!-\!\!\circ B \circ\!\!-\!\!\circ C$,$A \circ\!\!\rightarrow B \circ\!\!-\!\!\circ C$ or $A \circ\!\!-\!\!\circ B \leftarrow\!\!\circ C$) is considered a non-collider. Note that the definition simplifies somewhat for for example MVR CGs since they cannot contain any undirected edges. We also say that a collider resp. non-collider is shielded if $A$ and $C$ is adjacent, otherwise we say that it is unshielded.

The *independence model* $M$ induced by a graph $G$, denoted as $I(G)$ or $I_{PGM-class}(G)$, is the set of separation statements $X \perp_G Y|Z$ that hold in $G$ according to the interpretation to which $G$ belongs or the subscripted PGM-class. We say that two graphs $G$ and $H$ are *Markov equivalent* (under the same interpretation) or that they are in the same *Markov equivalence class* iff $I(G) = I(H)$. If an arrowhead occurs on an edge between the same nodes in all graphs of a Markov equivalence class we say that it is indifferent, otherwise we say that it is not indifferent. Moreover we also say that an edge is indifferent if it exists in every graph of a Markov equivalence class. Finally we do also define the skeleton of a graph $G$ as a graph with the same structure as $G$ with the exception that all edges have been replaced by undirected edges.

## Essential MVR CGs

As noted in section 3.1 an essential MVR CG $G^*$ of a MVR CG $G$ if it has the same skeleton as $G$ and contain all and only the arrowheads that are shared by all MVR CGs in the Markov equivalence class of $G$. Hence another definition can be that the essential MVR CG $G^*$ of a MVR CG $G$ is the JCG created when all MVR CGs in the Markov equivalence class of $G$ are joined.

We can now go on to prove Theorem 1:

*Proof.* From [25, Theorem 1] we know that any other MVR CG $G'$ in the same Markov equivalence class as $G$ must contain the same adjacencies as well as the same unshielded colliders. Since a collider over a node $Y$ between two nodes $X$ and $Z$ is a subgraph of the form $X \circ\!\!\rightarrow Y \leftarrow\!\!\circ Z$ with arrowheads towards $Y$ we know that these arrowheads must be in every MVR CG $G'$ in the same Markov equivalence class as $G$. Hence we know that the collider also must be in $G^*$. Since the definition of $d$-separation is the same for the JCGs and MVR CGs, and no new colliders can be created or removed when graphs of the same Markov equivalence class are joined, we know that the theorem must hold. □

For the proofs for the remainder of the theorems in subsection 3.1 we do however need to show how an essential MVR CG $G^*$ can be found from a MVR CG $G$. In their work of presenting essential graphs for AGs Ali et al. defines an algorithm for transforming an ancestral graph into its essential graph. Since MVR CGs are a subclass of AGs this algorithm do of course also work for creating the essential MVR CG of some MVR CG. However, just like for the characteristics some simplifications (and optimizations) can be made if one only considers the more restricted MVR CGs. The goal of the algorithm is to find all indifferent arrowheads in a graph, i.e. all the arrowheads that must exist in all graphs of that Markov equivalent class. This is exactly what is done in (parts of) the PC-algorithm and hence the MVR CG PC-algorithm [21] can be used to transform a MVR CG into its essential graph. If one studies the algorithm given by Ali et al. it can in fact also be seen that if it is simplified to only work for MVR CGs it becomes similar to the MVR CG PC-algorithm. We have here added the MVR-PC-algorithm (row 2 to 9) slightly modified to find the essential graph:

Transformation Algorithm

Given a MVR CG $G$ the algorithm learns a joined graph $G^*$ that is the essential graph of $G$

1 Let $G^*$ be the skeleton of $G$
2 Orient any induced subgraph $X \circ\!\!-\!\!\circ Y -\!\!\circ Z$ in $G^*$ to $X \circ\!\!\rightarrow Y \leftarrow\!\!\circ Z$ iff $X$ and $Z$ form an unshielded collider over $Y$ in $G$.
3 Apply the rules in Figure 1 onto $G^*$ until no further arrowheads are added.

**Algorithm 1:** Transformation algorithm

**Theorem 5.** *The transformation algorithm given in Algorithm 1 is correct and always learns the essential MVR CG $G^*$ given an MVR CG $G$ as input.*

*Proof.* To see that the transformation algorithm is correct we have to show two things. First, that the algorithm given in Algorithm 1 gives the same result as the first 9 lines of the PC-algorithm for MVR CGs [21] when we have a MVR CG $G$ instead of a probability distribution $p$ faithful to $G$ as input. Here we know that line 1 to 7 in the MVR CG PC-algorithm finds the skeleton of $G$, and hence, since we already have the skeleton of $G$, can be replaced by line 1 in the transformation algorithm shown here. We can also replace rule 0 in line 8 in the MVR CG PC-algorithm by line 2 in the transformation algorithm since we know that any node $B$ for which rule 0 can be applicable must be an unshielded collider in $G$. Finally line 9 in the MVR CG PC-algorithm can be replaced by line 3 here since we know that all unshielded colliders are found and orientated in $G^*$ in line 2 in the transformation algorithm. Hence for any triplet of nodes $A, B, C$, st the induced subgraph $A \circ\!\!-\!\!\circ B -\!\!\circ$ exists in $G^*$, $B$ must be in the separator of $A$ and $C$ if it reaches line 3. Hence this prerequisite can be removed from the rules, which are otherwise the same as for the MVR CG PC-algorithm.

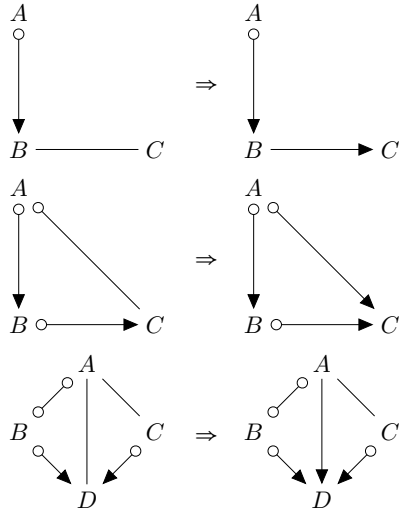



Figure 1: The rules

Secondly, we must show that first 9 lines of the MVR-PC algorithms gives the essential graph. This follows directly from that all the rules are sound [21, Lemma 3] and that any node can be chosen to be node of order 0 when orientating the remaining undirected edges in line 10 to 14. That the rules are sound means that the arrowheads introduced must exist in every MVR CG of the Markov equivalence class of $G$. Hence all arrowheads in $G^*$ after line 3 in Algorithm 1 are indifferent in the Markov equivalence class of $G$. That any node can be chosen to have order 0 means that any remaining undirected edge can be orientated as a directed edge in either direction and the resulting MVR CG can still belong to the same Markov equivalence class if the rest of the undirected edges are oriented appropriately. Hence all undirected edges $X$–$Y$ have at least one MVR CG $G_1$ st $I(G_1) = I(G)$ where the edge $X \to Y$ exists in $G_1$ and one MVR CG $G_2$ st $I(G_2) = I(G)$ where the edge $X \leftarrow Y$ exists in $G_2$. □

Having this transformation algorithm we can now define some characteristics of essential MVR CGs:

**Lemma 3.** *For any essential MVR CG $G^*$ we know that:*

1. *All bidirected edges are indifferent in the Markov equivalence class of MVR CGs that $G^*$ is the essential MVR CG for.*
2. *No undirected edge can share an endnode with a bidirected edge.*
3. *An induced subgraph of the form $A \to B$–$C$ cannot exist in an essential MVR CG. Hence all nodes in any undirected component must share the same parents.*

4. *Any undirected component is chordal.*
5. *No semi-directed cycles can occur.*

*Proof.* Point 1 follows directly from that all arrowheads are indifferent.

Point 2 follows from [21, Lemma 7].

Point 3 must hold or the first rule in Figure 1 would be applicable which is a contradiction.

Point 4 follows from [21, Lemma 8].

Point 5 must hold since we know that there exist a MVR CG $G$ with the same directed and bidirected edges as $G^*$ but where every undirected edge is made directed st $I(G) = I(G^*)$. Hence we know that no semi-directed cycle can occur in $G^*$ with only directed and bidirected edges since such a semi-directed cycle then would occur in $G$ which is a contradiction. On the other hand, if an undirected edge $X-Y$ is part of the semi-directed cycle we know that $pa_{G^*}(X) = pa_{G^*}(Y)$ by point 3 above, and hence there must also exist a semi-directed cycle without the undirected edge in $G^*$ because $X$ and $Y$ have no spouses by point 2 above. This reasoning can be repeated for every undirected edge in the semi-directed cycle. This means that if there existed a semi-directed cycle containing undirected edges there also must exist a semi-directed cycle without any undirected edges, which we know cannot be the case. Hence we have a contradiction. □

Which finally allows us to prove Theorem 2:

*Proof.* We will first prove that if any of the conditions are not fulfilled than the graph cannot be an essential MVR CG. Point 1 follows directly from point 5 in Lemma 3. Point 2 follows from the definition of indifferent arrowheads. Point 3 follows from point 4 in Lemma 3 and point 4 follows from point 2 and 3 in Lemma 3.

To show that the graph must be an essential MVR CG it is enough to show that if the conditions are fulfilled then an MVR CG exists in the same Markov equivalence class since we know that all arrowheads in $G$ are indifferent. Hence we need to show that the undirected edges in $G$ can be oriented as directed edges st no new unshielded colliders are added or semi-directed cycles are created. Since the undirected components are chordal we know that there exists a way to orient every edge as a directed edge st no semi-directed cycles or unshielded colliders are added including only the oriented edges [14, Theorem 4.13]. However, we still need to show that no new unshielded colliders or semi-directed cycles are created including the already directed edges in $G$. To see this note that no new unshielded colliders can be created since $pa_G(X) = pa_G(Y)$ and $sp_G(X) = sp_G(Y) = \varnothing$ for any two $X$ and $Y$ in the same undirected component. In addition, since all nodes in the same undirected component share the same parents it follows that if a semi-directed cycle, including a directed edge in $G$, would be created when the undirected edges are oriented, then there also must exist a semi-directed cycle in $G$. From point 1 we do however know that this is not the case, and

hence there must exist an MVR CG in the same Markov equivalence class as $G$ for which the undirected edges have been oriented. □

We can also prove Lemma 1 given in section 3:

*Proof.* Let $\mathcal{G}$ be the Markov equivalence class of $G^*$. That a DAG $G \in \mathcal{G}$ exists must hold since we know that there exist a MVR CG $G' \in \mathcal{G}$ with the same structure as $G^*$ with the exception that all undirected edges have been replaced by directed edges. Hence $G'$ only contains directed edges and hence is a DAG. That no DAG exists if $G^*$ contains any bidirected edge we simply have to note that DAGs is a subclass of essential MVR CGs and that all arrowheads are indifferent in an essential MVR CG. Hence there can exist no member in the Markov equivalence class not containing the bidirected edge.

To see that $G^*$ is the essential graph we can note the following. For any MVR CG $G' \in \mathcal{G}$ there must exist a DAG $G$ st all arrowheads in $G$ exist in $G'$ since $G'$ only can contain bidirected edges where $G$ contains directed edges. Hence it is enough to join every DAG in $\mathcal{G}$ to get the essential MVR CG $G^*$ and hence, since this is a way the essential graph can be created [4] for DAGs, $G^*$ must be the essential graph of $\mathcal{G}$. □

As well as Lemma 2:

*Proof.* Let $\mathcal{G}$ be the Markov equivalence class of $G^*$. We know that a covG $G \in \mathcal{G}$ must exist if there exists no non-shielded non-colliders in $G^*$ since for all triplets $X, Y$ and $Z$, st $G^*$ the induced subgraph $X \circ\!\!-\!\!\circ Y \circ\!\!-\!\!\circ Z$, $X$ and $Z$ must form an unshielded collider over $Z$. Hence all edges in $G^*$ can be replaced by bidirected edges to get $G$ st $I(G) = I(G^*)$. To see that no covG $G \in \mathcal{G}$ can exist if there exist a non-shielded non-collider in $G^*$ let this non-collider be over $Y$ st the induced subgraph $X \circ\!\!-\!\!\circ Y \circ\!\!-\!\!\rightarrow Z$ exists in $G^*$ st $X$ is a non-descendant of $Z$. Clearly $G$ must have the same adjacencies as $G^*$ since covGs is a subclass of essential MVR CGs and hence the induced subgraph $X \circ\!\!-\!\!\circ Y \circ\!\!-\!\!\circ Z$ must exist in $G$. However, since $G$ only contains bidirected edges this would mean that $X \not\perp_G Z | Y \cup pa_{G^*}(Z)$ while $X \perp_{G^*} Z | Y \cup pa_{G^*}(Z)$ must hold. Hence we have a contradiction. □

Finally we also state one more lemma about the structure of JCGs that is used later in the appendix.

**Lemma 4.** *The arrowhead on any directed edge that shares an endnode with an indifferent bidirected edge in a JCG must be indifferent itself.*

*Proof.* This follows from the fact that a bidirected edge cannot share an endnode with an undirected edge in an essential MVR CG as denoted in point 2 in Lemma 3. This means there can exist no directed edge $X \rightarrow Y$ in a MVR CG $G$ st there exist another MVR CG $G'$ of the same Markov equivalence class containing $X \leftarrow Y$ if $X$ or $Y$ is an endnode of a bidirected edge. Hence the arrowhead must be indifferent. □

## MCMC operators

The rest of the appendix is devoted to proving Theorem 3. This is performed by first giving the proof of the theorem which then uses lemmas defined later in the appendix. This structure does hopefully allow the reader to get a better understanding why the lemmas are needed and what they will be used for. Hence we first give the proof of Theorem 3:

*Proof.* Since we know that the possible states are the essential MVR CGs we need to prove two things; first that the Markov chain has a unique stationary distribution and secondly that this is the uniform distribution. The formed is proven if we show that the operators have the properties aperiodicity, i.e. that the Markov chain does not end up in the same state periodicly, and irreducibility, i.e. that any state can be reached from any other state. The latter is proven if we can show that the operators have the property reversibility, i.e. that the probability of transition from one state to another is equal to the probability of going to the latter state to the former.[11]

To prove aperiodicy we only need to show that there exists a positive probability of ending up in the same state in two consequent transitions (given that irreducibility holds) [11]. This obviously holds since there must always exist an operation that for a set of certain nodes will result in that $G_{i+1} = G_i$ (i.e. if "Add undirected edge" operation for two nodes $A$ and $B$ is possible, then the "Remove undirected edge" operation for the two nodes $A$ and $B$ cannot be possible).

To prove irreducibility we need to show that we with these operators can reach any essential MVR CG $G'$ from any other essential MVR CG $G''$. It is here enough to show that we from the empty graph $G_\varnothing$ can reach any essential MVR CG since we know that any operation is reversible. To show this we can base our reasoning on the following: let $G^*$ be the essential MVR CG we want to reach through a set of operations. It must then exist another MVR CG $G_u^*$ that has the same structure as $G^*$ with the exception that $G_u^*$ contains no undirected edge that exists in $G^*$. Lemma 5 states that we can remove edge by edge to reach $G_u^*$ from $G^*$ by only removing undirected edges one by one, and due to reversibility we then also know $G^*$ is reachable by adding undirected one by one to $G_u^*$. Hence we now only need to show that $G_u^*$ is reachable from the empty graph $G_\varnothing$. Lemma 7 then states that an essential MVR CG $G_{C_{i+1}}$ is reachable, with the operations described in Definition 2, from another essential MVR CG $G_{C_i}$ if $G_{C_{i+1}}$ and $G_{C_i}$ have the same structure with the exception that $G_{C_{i+1}}$ contains one more bidirected component $C$ st $ch_{G_{C_{i+1}}}(X) = \varnothing$. Hence we know that we can add components one by one st we achieve a set of essential MVR CGs $G_{C_0}, G_{C_1}, ..G_{C_n}$ where $G_{C_0} = G_\varnothing$ and $G_{C_n} = G_u^*$ and that all the necessary transformations is possible if the operations in Definition 2 is chosen in the right order. Hence we have irreducibility.

Finally we have reversibility. This must also hold since the probability of choosing an operation $o$ for an essential MVR CG $G_i$ that transforms $G_i$

to an essential MVR CG $G_{i+1}$ must be equal to choosing an operation $o'$ for $G_{i+1}$ st $o'$ transforms $G_{i+1}$ to $G_i$. To see this note that the probability of choosing any of the first six operators for a certain set of nodes is $\frac{1}{8} * \frac{1}{n^2}$ for any essential MVR CG. Hence the "remove" operator for a certain kind of edge must have an equal probability of being chosen in $G_{i+1}$ as the "add" operator for the same kind of edge in $G_i$ for these operators. For the seventh and eight operation the probability is harder to state. However, let $G_i$ be the essential MVR CG before a "add V-collider operation" over a node $X$, where $k$ non-arrow edge-endings have been changed to arrows, and $G_{i+1}$ the essential MVR CG after the operation. Then the probability of the "add V-collider operation" in $G_i$ is $\frac{1}{8} * \frac{1}{n} * \frac{1}{|ad_{G_i}(X)|^k}$. If we then study the reverse operator "Remove V-collider" in $G_{i+1}$ we can see that the probability of transforming $G_{i+1}$ to $G_i$ is $\frac{1}{8} * \frac{1}{n} * \frac{1}{|ad_{G_{i+1}}(X)|^k}$. Since we know that $ad_{G_i}(X) = ad_{G_{i+1}}(X)$ we can deduce that the probability must be equal in both cases. Hence we have reversibility for the set of operations. □

**Lemma 5.** *In a chordal undirected graph $G$, with at least one edge, it is always possible to remove one edge so that the resulting graph $G'$ is still chordal.*

*Proof.* From Jensen and Nielsen [16, Theorem 4.1] we know that there exists at least two simplicial nodes in $G$ such that, since $G$ contains an edge, have at least one neighbour. A simplicial node is a node st the set of neighbours of the node is complete. Let $X$ be a simplicial node that has a neighbour $Y$. Now assume that removing $X-Y$ from $G$ creates a nonchordal cycle. Then that cycle must be of the form $V_1, ..., V_n = V_1$ with $V_2 = X$ st $V_1$ and $V_3$ is not adjacent (or the cycle must also be non-chordal in $G$ which cannot be the case). However, that $V_1$ and $V_3$ is not adjacent is a contradiction since $X$ is simplicial, and hence $G \setminus \{X-Y\}$ must be chordal. □

**Lemma 6.** *In an essential MVR CG $G^*$ with a bidirected component $C$ there exists an unshielded collider, between two nodes of which at least one is in $C$ over every node in $C$.*

*Proof.* Assume the contrary. This means there exist a node $X \in C$ st no node in $C$ forms an unshielded collider over $X$. This does however mean there exists an essential MVR CG $G'$ with the same structure as $G^*$ with the exception that $X$ is a parent of $C$ instead of in it. This in turn means that $G'$ and $G^*$ represents the same Markov equivalence class but that $G'$ contains less arrowheads than $G^*$ which is a contradiction. □

**Lemma 7.** *Let $G^*$ and $G'$ be two essential MVR CGs without any undirected edges st $G^*$ and $G'$ have the same nodes and structure with the exception that $G'$ is missing a component $C$ that exists in $G^*$ st $ch_{G^*}(C) = \varnothing$. The operators in Definition 2 can then transform $G'$ to $G^*$ through a sequence of essential MVR CG $G_1, ..., G_n$ by only applying one operator at a time.*

*Proof.* If $|C| = 1$ this follows from 8, otherwise it follows from Lemma 9. □

Let $G^*$ and $G'$ be two essential MVR CGs without any undirected edges st $G^*$ and $G'$ have the same nodes and structure with the exception that $G'$ is missing a component $C$ containing only one node $Y$, that exists in $G^*$ st $ch_{G^*}(Y) = \emptyset$. The algorithm below then defines the operations and their order to transform $G'$ into $G^*$ through a sequence of essential MVR CGs $G_1, ..., G_n$. Let $G_i$ be the input graph for each line that is transformed into $G_{i+1}$, which then takes the place of $G_i$ in the next executed line:

1 Add $Y$ to $G'$ and denote the new graph $G_1$
2 Repeat until no case is applicable (restart the loop after each change):
3 If there exist a node $X \in pa_{G^*}(Y)$ st $X \notin pa_{G_i}(Y)$ and $sp_{G^*}(X) \neq \emptyset$ then let $G_{i+1} = G_i \cup \{X \to Y\}$.
4 If there exist a node $X \in pa_{G^*}(Y)$ st $X \notin pa_{G_i}(Y)$ and $bd_{G^*}(X) \notin pa_{G^*}(Y)$ then let $G_{i+1} = G_i \cup \{X \to Y\}$.
5 If there exist a node $X \in pa_{G^*}(Y)$ st $X \notin pa_{G_i}(Y)$ and $\exists Z \in pa_{G_i}(Y) \cap (co_{G_i}(X) \cup de_{G_i}(X))$ then let $G_{i+1} = G_i \cup \{X \to Y\}$.
6 If there exist a node $X \in pa_{G^*}(Y)$ st $X \notin pa_{G_i}(Y)$ and $pa_{G_i}(Y) \notin bd_{G^*}(X)$ then let $G_{i+1} = G_i \cup \{X \to Y\}$.
7 If there exist two nodes $X, Z \in pa_{G^*}(Y)$ st $X, Z \notin pa_{G_i}(Y)$ and $X \notin ad_{G^*}(Z)$ then let $G_{i+1} = G_i \cup \{X \to Y, Z \to Y\}$.

**Algorithm 2:** Procedure for adding edges when $|C| = 1$

**Lemma 8.** *All graphs $G_1, ..., G_n$ described in Algorithm 2 are essential MVR CGs and the transformation in each step can be achieved through one or more operations described in Definition 2 in which case all intermediate graphs are essential MVR CGs.*

*Proof.* Assume the contrary. This means that one of the following statements hold: (1) One of the lines in Algorithm 2 creates a graph $G_{i+1}$ that is not an essential MVR CG when $G_i$ is an essential MVR CG. (2) That $G_i$ is not transformable into $G_{i+1}$ through a sequence of operations described in Definition 2 or (3) that $G_i \neq G^*$ holds for the essential MVR CG $G_i$ that passes line 7. We will first show that (1) and (2) must be false and then finish the proof contradicting that (3) can hold. Note that since $G'$ is an essential MVR CG we know that all edges in the subgraph of $G_i$ induced by $V_{G_i} \setminus Y$ must be indifferent. This follows from the fact that the algorithm only add directed edges oriented towards $Y$ and hence that the cause making any arrowhead indifferent in $G'$ must still be valid in $G_i$. Also note that $G_i$ for any $i$ must be a subgraph of $G^*$ and hence that $ad_{G_i}(Y) = pa_{G_i}(Y)\ \forall i$.

We will first study $G_{i+1}$ for all lines in algorithm 2. To see that $G_{i+1}$ cannot contain any semi-directed cycle it is enough to note that $G_{i+1}$ is a subgraph of $G^*$ which can not contain any semi-directed cycles. Furthermore

any new arrowhead on a new edge in $G_{i+1}$ (compared to $G_i$) must be and remain indifferent in $G_j$ for all $j \geq i$ for the following reasons: line 3, since a directed edge that share an endnode with a bidirected edge always is indifferent according to Lemma 4. Line 4, since any other orientation of the edge would result in an unshielded collider not in $G^*$. Line 5, since any other orientation of the edge would cause a semi-directed cycle to appear in $G_j$, $j > i$. To see this note that all arrowheads in $G_i$ must be remain indifferent in all $G_j$ $j \geq i$. This can be seen inductively starting with $G_1$ which have the same edges as $G_i \setminus Y$ which must be indifferent for all $G_i$ as noted above. For each iteration $i$ of the algorithm one or more new directed edges is then added with the property that they will remain indifferent for all $j \geq i$. Hence all arrowheads in $G_i$ must remain indifferent for all $j \geq i$ and hence any other orientation of the edge added in line 5 must cause a semi-directed cycle in all $G_j$ $j \geq i$. Line 6, since an unshielded collider that exists in $G^*$ is added and line 7 since $X \to Y$ and $Z \to Y$ both are part of an unshielded collider in $G^*$. Hence $G_{i+1}$ must be essential MVR CGs for these lines.

In line 7 we will however have to do two or three operations to reach $G_{i+1}$ from $G_i$ and hence we have to show that all intermediate graphs are essential MVR CGs. Note that if this line is reached then (1) $bd_{G_i}(Y) \subseteq bd_{G_i}(X)$ (resp. $bd_{G_i}(Y) \subseteq bd_{G_i}(Z)$) and $sp_{G_i}(X) = sp_{G_i}(Z) = sp_{G_i}(Y) = \varnothing$ must hold. (1) must hold or line 6 must be applicable since $ad_{G_i}(X) \setminus Y = ad_{G^*}(X) \setminus Y$ (resp. $ad_{G_i}(Z) \setminus Y = ad_{G^*}(Z) \setminus Y$) and (2) must hold since we know that $sp_{G^*}(Y) = \varnothing$ and if $sp_{G_i}(X) \neq \varnothing$ (resp. $sp_{G_i}(Z) \neq \varnothing$) line 3 must be applicable.

We can now have two cases. If $bd_{G_i}(X) \notin bd_{G_i}(Y)$ (resp. $bd_{G_i}(Z) \notin bd_{G_i}(Y)$) then we know that $G_i \cup \{X \to Y\}$ (resp. $G_i \cup \{Z \to Y\}$) must be an essential MVR CG. Hence we can first add $X \to Y$ (resp. $Z \to Y$) to $G_i$ whereafter we can add $Z \to Y$ (resp. $X \to Y$) and then reach $G_{i+1}$. However if both $bd_{G_i}(X) \subseteq bd_{G_i}(Y)$ and $bd_{G_i}(Z) \subseteq bd_{G_i}(Y)$ hold then we know that $bd_{G_i}(X) = bd_{G_i}(Z) = bd_{G_i}(Y)$ must hold. Hence both $G_i \cup \{X\text{–}Y\}$ and $G_i \cup \{X\text{–}Y, Y\text{–}Y\}$ must be essential MVR CGs. We can then perform the "add V-collider" operation to transform the undirected edges to directed edges oriented towards $Y$ in $G_i \cup \{X\text{–}Y, Y\text{–}Y\}$ to reach $G_{i+1}$. Hence all intermediate graphs for line 7 must be essential MVR CGs.

The last thing to prove is then that $pa_{G_i}(Y) = pa_{G^*}(Y)$ after line 7. To see this assume the contrary, i.e. that there exist a node $X \in pa_{G^*}(Y)$ but $X \notin pa_{G_i}(Y)$. In addition, let us choose this $X$ st no node $W$ exists for which $W \in (de_{G^*}(X) \setminus co_{G^*}(X)) \cap pa_{G^*}(Y)$ and $W \notin pa_{G_i}(Y)$ hold. Obviously $sp_{G_i}(X) = \varnothing$ or line 3 would be applicable. We also know that $pa_{G^*}(X) = pa_{G_i}(X) \subseteq pa_{G^*}(Y)$ or line 4 would be applicable. $bd_{G^*}(Y) \subseteq ad_{G^*}(X)$ must also hold since no unshielded collider can exist between $X$ and some node $Z$ over $Y$ in $G^*$ or line 6 (if $Z \in pa_{G_i}(Y)$) or line 7 (if $Z \notin pa_{G_i}(Y)$) would be applicable. This means that $pa_{G^*}(X) = pa_{G^*}(Y) \setminus X$. Finally we also know that reversing the orientation of the edge $X \to Y$ in $G_i$ would cause no semi-

directed cycle to appear or line 5 would be applicable. This does however, since no node $W$ existed and hence that all descendants of $X$ already is added as parents to $Y$, that the orientation of the edge $X \to Y$ also can be reversed in $G^*$ without creating any semi-directed cycle or unshielded collider. This does however contradict that $G^*$ is an essential MVR CG. Hence we have a contradiction that $X$ can exists after line 7. □

Let $G^*$ and $G'$ be two essential MVR CGs without any undirected edges st $G^*$ and $G'$ have the same nodes and structure with the exception that $G'$ is missing a component $C$, st $|C| \geq 2$, that exists in $G^*$ st $ch_{G^*}(C) = \emptyset$. The algorithm below then defines the operations and their order to transform $G'$ into $G^*$ through a sequence of essential MVR CGs $G_1, ..., G_n$. Let $G_i$ be the input graph for each line that is transformed into $G_{i+1}$, which then takes the place of $G_i$ in the next executed line:

1 Add all the nodes in $C$ to $G'$ and denote the new graph $G_1$
2 Repeat until until $G_i = G^*$:
3 Let $X$, $Y$ and $Z$ be three nodes in $G_i$ st $X \in C, Y \in C$, $Z \in pa_{G^*}(C) \cup C$, $X$ and $Z$ form an unshielded collider over $Y$ in $G^*$, $bd_{G_i}(Y) = \emptyset$ and $\forall W \in ch_{G_i}(Y)$ st $C_w = co_{G_i}(W)$ there $\exists R, P \in C_w$ st $bd_{G_i}(R) \notin bd_{G^*}(Y) \cup Y$, $P \in ch_{G_i}(Y)$ and $bd_{G_{i+1}}(Y) \notin bd_{G^*}(P) \cup P$ hold. Let $G_{i+1} = G_i$ with the exception that all non-arrowheads towards $Y$ are replaced by arrowheads and if $X$ and/or $Z$ is not adjacent of $Y$ in $G_i$ then $X \to Y$ and/or $Z \to Y$ is added to $G_{i+1}$.
4 Repeat until $bd_{G_i}(Y) = bd_{G^*}(Y)$:
5 Let $Q$ be a node st $Q \in bd_{G^*}(Y)$ but $Q \notin bd_{G_i}(Y)$. Then let $G_{i+1} = G_i \cup \{Q \to Y\}$.

**Algorithm 3:** Procedure for adding edges when $|C| > 1$

**Lemma 9.** *All graphs $G_1, ..., G_n$ described in Algorithm 3 are essential MVR CGs and the transformation in each step can be achieved through one or more operations described in Definition 2 in which case all intermediate graphs are essential MVR CGs.*

*Proof.* Assume the contrary. This means that one of the following statements hold: (1) One of the lines in Algorithm 3 creates a graph $G_{i+1}$ that is not an essential MVR CG when $G_i$ is an essential MVR CG. (2) That $G_i$ is not transformable into $G_{i+1}$ through a set of operations described in Definition 2 or (3) that $G_i$ never becomes $G^*$. We also make two assumptions that have to be proven. Assumption 1 is that when when the loop in line 2 restarts we have that $\forall c_j \in C$, st $bd_{G_i}(c_j) \neq \emptyset$, $bd_{G_i}(c_j) = bd_{G^*}(c_j)$ must hold for the current graph $G_i$. We will denote these $c_j$ as being "collided". Assumption 2 is that all bidirected edges in any $G_i$ must be indifferent in all $G_j$ $j \geq i$.

The rest of the proof is constructed as follows. In part 1 we will first show that assumption 1 holds. This is then followed by proving that all $G_{i+1}$ are essential MVR CGs for line 3 in part 2. In the end of part 2 we also prove assumption 2. In part 3 we then show that $G_i$ is transformable into $G_{i+1}$ through a set of operations described in Definition 2 and that all intermediate graphs also are essential MVR CGs. In part 4 we show that $G_{i+1}$ must be an essential MVR CG for line 5 and finally in part 5 we show that the algorithm must terminate and hence that $G_i$ becomes $G^*$ after $|C|$ number of iterations of line 2.

First we will however make some observations about the algorithm and $G_i$. Note that only two lines, line 3 and line 5, changes the structure of $G_i$ to $G_{i+1}$. For these lines we can see that a directed edge $X \to Y$ only is added to $G_{i+1}$ if $X \to Y$ or $X \leftrightarrow Y$ is in $G^*$. We can also see that a bidirected edge $X \leftrightarrow Y$ only is added to $G_{i+1}$ in line 3 and then only if it also exists in $G^*$ since any $X \in ch_{G_i}(Y)$ and $Y$ must both be adjacent and in the same bidirected component in $G^*$. We can also note that edges never are removed once they have been added to $G_i$, although directed edges can be replaced by bidirected edges. This means that once an arrowhead is added to $G_{i+1}$ it must exist in all $G_j$ $j > i$. Finally we can also note that edges only are added with arrowheads towards collided nodes, i.e. nodes that has been chosen as $Y$ in line 3.

Part 1:

That $\forall c_j \in C$, st $bd_{G_i}(c_j) \neq \varnothing$, $bd_{G_i}(c_j) = bd_{G^*}(c_j)$ obviously holds for $i = 1$ since $bd_{G_1}(c_j) = \varnothing$ $\forall c_j \in C$. Then, for each iteration of line 2, one node $Y$ is selected in line 3 and becomes collided. When a node is chosen as $Y$ in line 3 two things happen; First, in line 3, all already adjacent nodes of $Y$ become spouses of $Y$ when all non-arrowheads towards $Y$ are replaced with arrowheads. Secondly, in line 5, that directed edges are added from all remaining nodes in $bd_{G^*}(Y)$ to $Y$. From this it is clear that once this is done all nodes in $bd_{G^*}(Y)$ also must be in $bd_{G_i}(Y)$.

We can also note that the algorithm never adds any edges oriented towards any other nodes than those chosen as $Y$, nor does it ever remove any arrowheads. This means that all nodes $c_j \in C$, for which $bd_{G_i}(c_j) \neq \varnothing$ holds, must have been chosen as $Y$ in some iteration $k$, and hence we know that $bd_{G_l}(c_j) = bd_{G^*}(c_j)$ must hold $\forall l > k$. Hence it is clear from induction, starting with $i = 1$ that $\forall c_j \in C$, st $bd_{G_i}(c_j) \neq \varnothing$, $bd_{G_i}(c_j) = bd_{G^*}(c_j)$ must hold $\forall G_i$ hence that assumption 1 holds.

Part 2:

In this part we study line 3 and show that all $G_{i+1}$ are essential MVR CGs for line 3 and in the end we also show that assumption 2 holds. Here we have three nodes $X \in C$, $Y \in C$ and $Z \in pa_{G^*}(C) \cup C$ st $X$ and $Z$ forms an unshielded collider over $Y$ in $G^*$. We also know that $bd_{G_i}(Y) = \varnothing$ and $\forall W \in ch_{G_i}(Y)$ st $C_w = co_{G_i}(W)$ there $\exists R, P \in C_w$ st $bd_{G_i}(R) \notin bd_{G^*}(Y) \cup Y$, $P \in ch_{G_i}(Y)$ and $bd_{G_{i+1}}(Y) \notin bd_{G^*}(P) \cup P$ hold.

We can also make some deductions about the structure of $G_i$. Recall

that $ch_{G^*} = \emptyset$ and hence we know, together with assumption 1, that no collided nodes in $C$ can have any children. From assumption 1 we also know that any child of $Y$, as well as any node in $C$ that is endnode of a bidirected edge, must be collided. This in turn means that $\forall c_j \in C$ st $bd_{G_i}(c_j) \neq \emptyset$, we must have that $ch_{G_i}(c_j) = \emptyset$ which means that $de_{G_i}(c_j) \setminus co_{G_i}(c_j) = \emptyset$ must hold.

Also note that we can have three subcases: Case 1, that both $X, Z \in ch_{G_i}(Y)$, case 2, that $X \in ch_{G_i}(Y)$ but $Z \notin ad_{G_i}(Y)$ and finally case 3, that $X, Z \notin ad_{G_i}(Y)$. If we have case 1 line 3 then consists of replacing all non-arrowhead edgeendings towards $Y$ with an arrowheads simultaneously. Hence all children of $Y$ will become spouses of $Y$. In case 2 all non-arrowheads are replaced for $G_{i+1}$ but $Z \to Y$ is also added. Finally in case 3 both $X \to Y$ and $Z \to Y$ is added to $G_{i+1}$ in addition to the replacement of the non-arrowheads.

We can now prove that $G_{i+1}$ cannot contain any semi-directed cycle since we know that $bd_{G_i}(Y) = \emptyset$ and that $\forall c_j \in ch_{G_i}(Y)$ we must have that $de_{G_i}(c_j) \setminus co_{G_i}(c_j) = \emptyset$. If we assume case 1 we know that no new directed edges are added and hence that $\forall c_j \in ch_{G_i}(Y)$ $ch_{G_i}(c_j) = ch_{G_{i+1}}(c_j)$. We also know that $ch_{G_{i+1}}(Y) = \emptyset$ and, since $\forall c_j \in co_{G_{i+1}}(Y) \cup Y$ $ch_{G_{i+1}}(c_j) = \emptyset$, we obviously cannot have a semi-directed cycle. For case 2 we can note that we can have two cases, either $Z \in C$ or $Z \notin C$. If $Z \in C$ then we know that $Z$ cannot be previously collided or $Z \in ch_{G_i}(Y)$ would have to hold due to assumption 1 and that $Z \in sp_{G^*}(Y)$. Hence we know that $bd_{G_i}(Z) = \emptyset$ and since the only edge containing $Z$ as an endnode that is added in line 3 is $Z \to Y$ and hence $bd_{G_{i+1}}(Z) = \emptyset$ must hold. This does however contradict that $Z$ is part of any semi-directed cycle. If we on the other hand have that $Z \notin C$ then we know that $\forall c_j \in C$ $Z \notin de_{G_i}(c_j)$ and $Z \notin de_{G_{i+1}}(c_j)$ must hold. Hence we cannot have a semi-directed cycle containing $Z$ in $G_{i+1}$ for case 2 and that no semi-directed cycle can exist not containing $Z$ follows similarly as for case 1. For case 3 we can see that no semi-directed cycle can exist in $G_{i+1}$ and contain $Z$ or $X$ similarly as we saw in case 2 since $X$ must be in $C$. It also follows similarly as for case 1 that no semi-directed cycle can exist in $G_{i+1}$ containing neither $X$ nor $Z$. Hence no semi-directed cycle can exist in $G_{i+1}$.

Now assume that a not indifferent arrowhead is created in $G_{i+1}$ when the non-arrowheads are replaced by arrowheads. We know that the edge $X \to Y$ (resp. $Z \to Y$), for case 2 or 3 (resp. case 3) must be part of an unshielded collider over $Y$ and hence be indifferent. This means that the not indifferent arrowhead must be on one of the bidirected edges created containing $Y$ as an endnode. Let $S$ be the node for which either the arrowhead towards $Y$ or towards $S$, or both, on the edge $S \leftrightarrow Y$ is not indifferent in $G_{i+1}$. We can then note that $S$, as well as all nodes in $C_S = co_{G_i}(S)$, must be collided previously since $Y \to S$ is in $G_i$ due to assumption 1. From the prerequisite of choosing $Y$ in line 3 we know that there $\exists c_n, d_m \in C_S$ st $bd_{G_i}(c_n) \notin bd_{G^*}(Y) \cup Y$, $d_m \in ch_{G_i}(Y)$ and $bd_{G_{i+1}}(Y) \notin bd_{G^*}(d_m) \cup d_m$

hold. Note that $c_n$ and $d_m$ can be $S$. Let $S \leftrightarrow c_1 \leftrightarrow c_2 \leftrightarrow ... \leftrightarrow c_n$, $n \geq 0$ (0 if $c_n = S$), be a path $\pi_1$ between $S$ and $c_n$ in $G_i$. Without losing generalization we can assume that $c_n$ must be adjacent of $Y$ or there must exist some node $c_k \in S \cup c_1 \cup ... \cup c_{n-1}$ st $c_k$ is adjacent of $Y$ and st $bd_{G_i}(c_k) \notin bd_{G^*}(Y) \cup Y$ and hence can take the place of $c_n$. Hence we know that $Y \leftrightarrow c_n$ must be in $G_{i+1}$. From assumption 2 we then know that all bidirected edges on $\pi_1$ must be indifferent in $G_{i+1}$ since they also existed in $G_i$. From $bd_{G_i}(c_n) \notin bd_{G^*}(Y) \cup Y$ we then know that the arrowhead towards $c_n$ on $Y \leftrightarrow c_n$ must be indifferent in $G_{i+1}$ since it is part of an unshielded collider in $G_i$ and $bd_{G_{i+1}}(c_n) = bd_{G_i}(c_n)$. This in turn means that the arrowhead towards $S$ on $Y \leftrightarrow S$ must be indifferent or a semi-directed cycle would exist in $G_{i+1}$. Similarly we can see that there must exist a path $\pi_2$ of indifferent bidirected edges between $S$ and $d_m$ and that the arrowhead towards $Y$ on $Y \leftrightarrow d_m$ must be indifferent in $G_{i+1}$. This is due to that $bd_{G_{i+1}}(Y) \notin bd_{G^*}(d_m) \cup d_m$ holds according to the prerequisite and $bd_{G^*}(d_m) = bd_{G_i}(d_m)$ since $d_m$ has been collided previously and hence that $d_m$ forms an unshielded collider over $Y$ with some other node in $G_{i+1}$. This in turn means that the arrowhead towards $Y$ on $Y \leftrightarrow S$ must be indifferent or a semi-directed cycle occurs in $G_{i+1}$. Hence we have shown that the arrowhead towards $S$ is indifferent due to $c_n$ and that the arrowhead towards $Y$ is indifferent due to $d_m$ on the edge $S \leftrightarrow Y$ and contradicted that any arrowhead on this edge can be indifferent. Hence all arrowheads added in line 3 must be indifferent.

Furthermore we can also see that the reasoning must hold for any edge $S \leftrightarrow Y$ for all $G_j$ $j > i$ because of the following: (1) the arrowhead towards $c_n$ on $c_n \leftrightarrow Y$ must be indifferent in all $G_j$ since $bd_{G_i}(c_n) = bd_{G^*}(c_n)$ must hold since $c_n$ has been collided previously. This then means that $bd_{G_j}(c_n) \notin bd_{G^*}(Y) \cup Y$ must hold for all $j > i$. (2) the arrowhead towards $Y$ on $d_m \leftrightarrow Y$ must be indifferent in all $G_j$ since $bd_{G_{i+1}}(Y) \subseteq bd_{G_j}(Y)$ and hence $bd_{G_j}(Y) \notin bd_{G^*}(d_m) \cup d_m$ must hold for all $j$. (3) Finally we can show that all bidirected edges in $co_{G_i}(S)$ must be indifferent for all $G_j$ $j > i$ iteratively. Starting with $G_1$ containing no edges with endnodes in $C$ this statement obviously holds, and each time a bidirected edge is added to $G_i$ it must hold that all previously added bidirected edges in $G_i$ must be indifferent for all $G_j$ $j \geq i$. Therefore, following the reasoning above, the bidirected edges between any nodes in $sp_{G_{i+1}}(Y)$ and $Y$ must also be indifferent for all $G_j$ $j > i$ and we have proven that assumption 2 is correct since this is the only place in the algorithm where bidirected edges are created.

What remains to prove to show that $G_{i+1}$ is an essential MVR CG is that no arrowhead are made not indifferent in line 3. Due to assumption 2 we then know that the edge made not indifferent cannot be a bidirected edge, and since $G_i$ contains no undirected edges, we know that it must be a directed edge. For an directed edge to have an indifferent arrowhead it must either be part of an unshielded collider, or the other orientation of it must cause a new unshielded collider or a semi-directed cycle to occur in the graph. Hence, since edges only are added and made bidirected in line 4, for an edge

to be made not indifferent it must cease to be part of an unshielded collider. This means that we can, without losing generality, assume a directed edge with an endnode $M$ cease to be part of an unshielded collider when $X \to Y$ is added to $G_i$. Then we know that $X \to M$ and $Y \to M$ also must exist in $G_i$ since $X$ and $Y$ must form an unshielded collider over $M$ in $G_i$ but neither $X$ nor $Y$ have been collided previously. This does however mean that $Y \leftrightarrow M$ must exist in $G_{i+1}$ which together with Lemma 4 contradicts that any adjacent directed edge can be not indifferent. Hence we have a contradiction and all arrowheads in $G_{i+1}$ must be indifferent which means that $G_{i+1}$ is an essential MVR CG for line 3.

Part 3

It is easy to see that $G_i$ is transformable into $G_{i+1}$ in line 5 since the transformation only consists of adding a directed edge and hence that operation can be used. For line 3 it is however more difficult. In part 2 three subcases, case 1, 2 and 3, were identified. If we have case 1 then it is easy to see that the transformation from $G_i$ to $G_{i+1}$ can be achieved by the "add V-collider" operation.

For case 2 we can note three things; first that $bd_{G_i}(Y) = \varnothing$, secondly that $X$ is collided and hence that $Y \to X$ is in $G_i$. Thirdly we can note that we can have two major sub-cases, either $bd_{G_i}(Z) = \varnothing$ or $bd_{G_i}(Z) \neq \varnothing$. First assume that $bd_{G_i}(Z) = \varnothing$ holds. We can then add an undirected edge between $Z$ and $Y$. The resulting graph $G_i \cup \{Z\text{–}Y\}$ must be an essential MVR CGs since $bd_{G_i}(Y) = bd_{G_i}(Z) = \varnothing$ and since $G_i$ did not contain any undirected edges the created undirected component is chordal. To see that no indifferent arrowheads are made not indifferent assume the contrary. Then there must exist a third node $M$ st $Y \to M$ and $Z \to M$ are in $G_i$ for which either the arrowhead on $Z \to M$ or $Y \to M$ is made indifferent similarly as we saw in part 2. From assumption 1 we know that $bd_{G_i}(M) = bd_{G^*}(M)$ and from Lemma 6 we know that there must exist an unshielded collider over $M$. Hence we know that an unshielded collider exists over $M$ in $G_i$. Note that $Z$ and $Y$ cannot be the two nodes forming the unshielded collider over $M$ since $Z$ is adjacent of $Y$ in $G^*$. Let $U$ be one of the nodes in $bd_{G^*}(M) \setminus (Z \cup Y)$ forming such an unshielded collider. If $U \in pa_{G^*}(C)$ or if $U \in C$ but has not been collided previously then there can exist no edge between $U$ and $Y$ since $bd_{G_i}(Y) = \varnothing$ and $bd_{G_i}(U) = \varnothing$. Hence $Y \to M$ must form an unshielded collider with $U$ over $M$ in $G_{i+1}$ which gives us a contradiction. If $U \in C$ and $U$ has been collided we know that $U \leftrightarrow M$ must be in $G_i$ and from Lemma 4 we then have that any directed edge sharing an endnode $M$ with an indifferent bidirected edge must be indifferent. Hence we have a contradiction and $G_i \cup \{Z\text{–}Y\}$ must be an essential MVR CG. From $G_i \cup \{Z\text{–}Y\}$ we can then perform the "add V-collider" operation to reach $G_{i+1}$.

If on the other hand $bd_{G_i}(Z) \neq \varnothing$ we know that $Z$ cannot be in $C$ or $Y \to Z$ would exist in $G_i$ according to assumption 1. We also know that $G_i \cup \{Z \to Y\}$ must be an essential MVR CG since reversing the edge

would cause the graph to have a different set of unshielded colliders. To see this note that $bd_{G_i}(Y) = \varnothing$ and that $nb_{G_i}(Z) = \varnothing$. Hence the edge $Y \to Z$ would cause $Y$ to form unshielded colliders over $Z$ with all nodes in $bd_{G_i}(Z)$ and hence the edge $Z \to Y$ must be indifferent. Also note that adding $Z \to Y$ cannot cause some already existing arrowhead in $G_i$ to become not indifferent according to the same reasoning as for $Z$–$Y$ above. Hence $G_i \cup \{Z \to Y\}$ must be an essential MVR CG.

For case 3 we can follow the same reasoning. In this case we do however that $X$ cannot have been collided previously and hence that $bd_{G_i}(X) = \varnothing$ must hold due to assumption 1. First assume that $bd_{G_i}(Z) = \varnothing$ holds. We can then add undirected edges, first between $X$ and $Y$ and then between $Z$ and $Y$. The resulting graphs $G_i \cup \{X$–$Y\}$ and $G_i \cup \{X$–$Y, Z$–$Y\}$ must be essential MVR CGs since $bd_{G_i}(X) = bd_{G_i}(Y) = bd_{G_i}(Z) = \varnothing$ and since $G_i$ did not contain any undirected edges the created undirected component must be chordal. That no indifferent arrowheads are made not indifferent follows as for case 2 above. From $G_i \cup \{Z$–$Y \cup X$–$Y\}$ the "add V-collider" operation can then be performed to reach $G_{i+1}$.

If on the other hand $bd_{G_i}(Z) \neq \varnothing$ it follows similarly as for case 2 that $G_i \cup \{Z \to Y\}$ is an essential MVR CG. From this graph the edge $X \to Y$ can then be added which now will form an unshielded collider, since $bd_{G_i}(X) = \varnothing$, with $Z$ over $Y$ and hence $G_i \cup \{Z \to Y \cup X \to Y\}$ must be an essential MVR CG. Note that adding $Z \to Y$ (resp. $X \to Y$) cannot cause some already existing arrowhead in $G_i$ to become not indifferent according to the same reasoning as for $X$–$Y$ above.

This gives us that $G_i$ is transformable into $G_{i+1}$ using the operators in Definition 2 for all lines in Algorithm 3 and that all intermediate graphs are essential MVR CGs.

Part 4

In this part we will show that $G_{i+1}$ in line 5 must be an essential MVR CG if $G_i$ is an essential MVR CG for the same line. Line 5 add a directed edge $Q \to Y$ to $G_i$ for which we know that $\forall c_k \in co_{G_i}(Y)$ $c_k$ must be collided, due to assumption 1, and hence that $ch_{G_i}(c_k) = \varnothing$ similarly as we saw in part 2. Hence we know that $de_{G_i}(Y) \setminus co_{G_i}(Y) = de_{G_{i+1}}(Y) \setminus co_{G_{i+1}}(Y) = \varnothing$. From this it directly follows that $G_{i+1}$ can contain no semi-directed cycle since $G_i$ contains no semi-directed cycle.

Now assume that the arrowhead on the edge $Q \to Y$ is not indifferent in $G_{i+1}$. We know there exists a node $X$ as described in line 3 in the algorithm st $X \in bd_{G_i}(Y) \cap C$. Since $Y \leftrightarrow Q$ is not in $G_i$ we must also have that either $Q \notin C$ holds or that $Q$ has not been collided previously. In either case we know that $Q \notin ch_{G_i}(X)$ must hold. First assume that $X$ has not been collided previously. Then, since $bd_{G_i}(X) = \varnothing$, $X$ and $Q$ cannot be adjacent and therefore must form an unshielded collider over $Y$ and hence the arrowhead on the edge $Q \to Y$ must be indifferent in $G_{i+1}$. If $X$ on the other hand has been collided previously we know that both $G_i$ and $G_{i+1}$ must contain the bidirected edge $X \leftrightarrow Y$. From Lemma 4 we then get that

any directed edge with $X$ or $Y$ as an endnode also must be indifferent and hence we have a contradiction that $Q \rightarrow Y$ is not indifferent in $G_{i+1}$.

Secondly assume that the edge $Q \rightarrow Y$ causes some other previously indifferent arrowhead to become not indifferent in $G_{i+1}$. Similarly as we saw in the last paragraph of part 2 we know that this must include a third node $M \in ad_{G_i}(Q) \cap ad_{G_i}(Y)$ st there is either the edge between $Q$ and $M$ or $Y$ and $M$ that is made not indifferent. We can from this note that the edge between $Y$ and $M$ cannot be bidirected since Lemma 4, together with assumption 2, then states that all directed edges containing $M$ as an endnode must be indifferent. Hence either $Y \rightarrow M$ or $M \rightarrow Y$ must exist in $G_i$. However, since only directed edges oriented towards $Y$ is added after line 3, and every child of $Y$ is made a spouse of $Y$ in line 3, we can see that the edge $Y \rightarrow M$ cannot exist in $G_i$ in line 5. Hence $M \rightarrow Y$ must exist in $G_i$. From this it follows that, since $Y$ and $X$ cannot form an unshielded collider over $M$ in $G_i$ when $M \rightarrow Y$ is in $G_i$, it must be the edge $M \rightarrow Y$ that is made not indifferent when $Q \rightarrow Y$ is added. However, just like in the last paragraph we can note that a node $X$ must exist, and that if $X$ is collided then $M \rightarrow Y$ must be indifferent due to the bidirected edge, while if it is not collided, then $M$ and $X$ must form an unshielded collider over $Y$ in $G_{i+1}$. Hence the edge $M \rightarrow Y$ must be indifferent in $G_{i+1}$ and we have a contradiction which means that $G_{i+1}$ for line 5 must be an essential MVR CG if $G_i$ is an essential MVR CG.

Part 5

The last thing to prove is that the algorithm must terminate and hence that $G_i$ becomes $G^*$ after $|C|$ number of iterations of line 2.

To show this assume the contrary, i.e. that there exist no node in $G_i$ for which line 3 is applicable but where there exist a node $Y$ that has not yet been collided and hence $bd_{G_i}(Y) \neq bd_{G^*}(Y)$. From Lemma 6 we know that an unshielded collider must exist over $Y$ in $G^*$ for which at least one of the boundary nodes are in $C$. From assumption 1 we also know that $\forall c_j \in C$, st $bd_{G_i}(c_j) \neq \emptyset$, $bd_{G_i}(c_j) = bd_{G^*}(c_j)$ must hold for $G_i$ and hence that $bd_{G_i}(Y) = \emptyset$ must hold. It is now easy to see that $ch_{G_i}(Y) \neq \emptyset$ must hold or line 3 would be applicable. In general this also means that for all non-collided nodes $c_l \in C$ we must have that $ch_{G_i}(c_l) \neq \emptyset$ or line 3 would be applicable for that node.

Before we continue we will note two things (1) First that for every connected set of nodes $A \subset C$, i.e. every $A \subset C$ for which there exist a path between any pair of nodes in the subgraph of $G^*$ induced by A, we must have that there $\exists a_l \in A$ st $\exists R \in sp_{G^*}(a_l)$ and $bd_{G^*}(a_l) \notin bd_{G^*}(R) \cup R$. If this would not be the case there would exist no unshielded collider over $A$. Hence there would exist a MVR CG $G'$, st $I(G') = I(G^*)$, where $G'$ has the same structure as $G^*$ with the exception that $A$ is a parentcomponent of $C \setminus A$ in $G'$ instead of in the same component as $C \setminus A$ as in $G^*$. This is of course a contradiction since $G'$ would contain less arrowheads than $G^*$ and hence $G^*$ cannot be an essential MVR CG. Similarly we can note (2)

that $\exists a_m \in A$ st $\exists P \in sp_{G^*}(a_m)$ and $bd_{G^*}(P) \notin bd_{G^*}(a_m) \cup a_m$ or $A$ could be a childcomponent of $C \setminus A$ and the graph would still represent the same independence model.

Let $B \subset C$ st $B$ forms a component in $G_i$. Obviously such a $B$ must exist or $C$ would form one large component and hence $G_i = G^*$. From (2) we know there exists at least one node $b_r \in B$ st there $\exists T \in pa_{G_i}(b_r) \cap C$ st $bd_{G^*}(T) \notin bd_{G^*}(b_r) \cup b_r$. To see this let $A$ in (2) take the form of $B$. Then we know there exists a node $a_m \in A$ st $\exists P \in sp_{G^*}(a_m)$ st $bd_{G^*}(P) \notin bd_{G^*}(a_m) \cup a_m$. Obviously $P \in sp_{G_i}(a_m)$ cannot hold, since $P$ then would belong to $B$, but since we know that $\forall b_k \in B$ $bd_{G_i}(b_k) = bd_{G^*}(b_k)$ by assumption 1, and that $P \in bd_{G^*}(a_m)$, we can deduce that $P \in pa_{G_i}(a_m) \cup nb_{G_i}(a_m)$ must hold. This together with the fact that $G_i$ does not contain any undirected edges means that $P \in pa_{G_i}(a_m)$. Hence there must exist a $b_r$ taking the form of $a_m$ and $T$ taking the form of $P$. Also note that if different nodes in $ch_{G_i}(T)$ belong to different components in $G_i$, then there must exist a node with the same properties as $b_r$ in all components. To see this note that $ch_{G_i}(T) \subseteq sp_{G^*}(T)$ and that all nodes in $ch_{G_i}(T)$ must have been collided according to assumption 1. For the prerequisites of line 3 not to be fulfilled we know that for some component $D$ that contains a child of $T$ we must have that $\forall d_k \in D$ $bd_{G_i}(d_k) \subseteq bd_{G^*}(T) \cup T$ must hold. Moreover, since $d_k$ must be collided we must also have that $bd_{G_i}(d_k) = bd_{G^*}(d_k)$ and hence that $bd_{G^*}(d_k) \subseteq bd_{G^*}(T) \cup T$ must hold. This in turn also means that $\forall d_k \in D$ $d_k \in ch_{G_i}(T)$ must hold. If we let $E = pa_{G_i}(D) \cap C$ and $F$ consist of all nodes $e_m \in E$ st $\exists d_n \in D$ and $bd_{G^*}(e_m) \notin bd_{G^*}(d_n) \cup d_n$ we can note that $\forall f_m \in F$ we must have that $\forall d_k \in D$ $bd_{G_i}(d_k) \subseteq bd_{G^*}(f_m) \cup f_m$ and $d_k \in ch_{G_i}(f_m)$ or the prerequisite for line 3 must be fulfilled for a node $f_m$. In addition, since $d_k$ must be collided in $G_i$ we must have that $bd_{G^*}(d_k) \subseteq bd_{G^*}(f_m) \cup f_m$ holds. This does however mean that $F$ is complete and that $\forall f_m \in F$ $D \cup (E \setminus F) \subseteq ad_{G^*}(f_m)$ and $ad_{G^*}(D \cup (E \setminus F)) \subseteq ad_{G^*}(f_m) \cup f_m$. This means that there exists no colliders towards any node in $D \cup (E \setminus F)$ from any other node in $C$. Hence we have a contradiction with (1) since there then would exist an essential MVR CG $G'$ with the same structure as $G^*$ with the exception that $D \cup (E \setminus F)$ are parents of $C$ instead of in $C$ which contradicts that $G^*$ is an essential MVR CG. Hence at least one of the nodes in $F$ must be possible to collide which contradicts the assumption. □

**Acknowledgments**

This work is funded by the Center for Industrial Information Technology (CENIIT) and a so-called career contract at Linköping University, by the Swedish Research Council (ref. 2010-4808), and by FEDER funds and the Spanish Government (MICINN) through the project TIN2010-20900-C04-03.

# References

[1] The On-Line Encyclopedia of Integer Sequences. Published Electronically http://oeis.org, 2010.

[2] R. A. Ali, T. S. Richardson, P. Spirtes, and J. Zhang. Towards Characterizing Markov Equivalence Classes for Directed Acyclic Graphs with Latent Variables. In *Proceedings for the 21st Conference on Uncertainty in Artificial Intelligence*, pages 10–17, 2005.

[3] S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of Markov Equivalence Classes for Acyclic Digraphs. *The Annals of Statistics*, 25:505–541, 1997.

[4] S. A. Andersson, D. Madigan, and M. D. Perlman. On the Markov Equivalence of Chain Graphs, Undirected Graphs and Acyclic Digraphs. *Scandianavian Journal of Statistics*, 24:81–102, 1997.

[5] S. A. Andersson, D. Madigan, and M. D. Perlman. An Alternative Markov property for Chain Graphs. *Scandianavian Journal of Statistics*, 28:33–85, 2001.

[6] R. Castelo. *The Discrete Acyclic Digraph Markov Model in Data Mining*. PhD thesis, Utrecht University, 2002.

[7] D. R. Cox and N. Wermuth. Linear Dependencies Represented by Chain Graphs. *Statistical Science*, 8:204–283, 1993.

[8] M. Frydenberg. The Chain Graph Markov Property. *Scandinavian Journal of Statistics*, 17:333–353, 1990.

[9] S. B. Gillispie. Formulas for Counting Acyclic Digraph Markov Equivalence Classes. *Journal of Statistical Planning and Inference*, pages 1410–1432, 2006.

[10] S. B. Gillispie and M.D. Perlman. The Size Distribution for Markov Equivalence Classes of Acyclic Digraph Models. *Artificial Intelligence*, pages 137–155, 2002.

[11] O. Häggström. *Finite Markov Chains and Algorithmic Applications*. Campbridge University Press, 2002.

[12] F. Harary and E. M. Palmer. *Graphical Enumeration*. Academic Press, 1973.

[13] Y. He, J. Jia, and B. Yu. Reversible MCMC on Markov Equivalence Classes of Sparse Directed Acyclic Graphs. arXiv:1209.5860v2 [stat.ML], 2012.

[14] D. Koller and N. Friedman. *Probabilistic Graphcal Models*. MIT Press, 1999.

[15] S. L. Lauritzen and N. Wermuth. Graphical Models for Association Between Variables, Some of Which are Qualitative and Some Quantitative. *The Annals of Statistics*, 17:31–57, 1989.

[16] T. D. Nielsen and F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2007.

[17] J. M. Peña. Approximate Counting of Graphical Models Via MCMC. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 352–359, 2007.

[18] M. D. Perlman. Graphical Model Search Via Essential Graphs. Technical report, University of Washington, Technical Report 367, 2000.

[19] R. W. Robinson. Counting Labeled Acyclic Digraphs. *New Directions in the Theory of Graphs*, pages 239–273, 1973.

[20] R. W. Robinson. Counting Unlabeled Acyclic Digraphs. In *Proceedings of the Fifth Australian Conference on Combinatorial Mathematics*, pages 28–43, 1977.

[21] D. Sonntag and J. M. Peña. Learning Multivariate Regression Chain Graphs under Faithfulness. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 299–306, 2012.

[22] D. Sonntag and J. M. Peña. Chain Graph Interpretations and Their Relations. In *Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, pages 510–521, 2013.

[23] B. Steinsky. Enumeration of Labelled Chain Graphs and Labelled Essential Directed Acyclic Graphs. *Discrete Mathematics*, pages 266–277, 2003.

[24] M. Volf and M. Studený. A Graphical Characterization of the Largest Chain Graphs. *International Journal of Approximate Reasoning*, 20:209–236, 1999.

[25] N. Wermuth and K. Sadeghi. Sequences of Regression and Their Independences. arXiv:1103.2523 [stat.ME], 2012.

[26] E. M. Wright. A Relationship between Two Sequences. In *Proceedings of the London Mathematical Society*, pages 296–304, 1967.

## 5.5 Article 4: Learning multivariate regression chain graphs under faithfulness

### Bibliography:

Title: Learning Multivariate Regression Chain Graphs under Faithfulness
Authors: Dag Sonntag and Jose M. Peña
Publication: In Proceedings of the 6th European Workshop on Probabilistic Graphical Models (2012)
Status: Accepted paper

# Learning Multivariate Regression Chain Graphs under Faithfulness

Dag Sonntag and Jose M. Peña

ADIT, IDA, Linköping University, Sweden

**Abstract** This paper deals with multivariate regression chain graphs, which were introduced by Cox and Wermuth (1993, 1996) to represent linear causal models with correlated errors. Specifically, we present a constraint based algorithm for learning a chain graph a given probability distribution is faithful to. We also show that for each Markov equivalence class of multivariate regression chain graphs there exists a set of chain graphs with a unique minimal set of lines. Finally, we show that this set of lines can be identified from any member of the class by repeatedly splitting its connectivity components according to certain conditions.
**Keywords:** Chain Graph, Multivariate Regression Chain Graph, Learning, Bidirected Graph

## 1 Introduction

In this paper we deal with multivariate regression chain graphs (CGs) which were introduced by Cox and Wermuth [2, 3]. Graphically Cox and Wermuth represent these CGs with dashed edges to distinguish them from other interpretations, e.g. LWF [6] or AMP [1]. Multivariate regression CGs also coincide with the acyclic directed mixed graphs without semi-directed cycles presented by Richardson [11]. A fourth interpretation of CGs can also be found in Drton (2009). The different interpretations of CGs have different merits, but none of the interpretations subsumes another interpretation [4].

The multivariate regression CG interpretation still misses some fundamental elements found and proven in the LWF and AMP interpretations. In this article we study and describe two of these elements. The first is a learning algorithm for learning a CG from a probability distribution faithful to a multivariate regression CG. Similar algorithms have been presented for the LWF [7, 13] as well as the AMP [10] interpretations. The algorithm presented is constraint based and resembles both Studený's and Peña's algorithms. The second element we present is a feasible split operation and a feasible merge operation similar to the ones presented for the LWF and AMP interpretations [14]. These splits and mergings can be used to alter the structure of a multivariate regression CG in such a way that it does not change the CG's Markov equivalence class. Finally we show that for each Markov equivalence class of multivariate regression CGs there exists a set of

CGs with a unique minimal set of lines. We also show that this set of CGs can be reached by applying feasible splits to any CG in the same Markov equivalence class.

The rest of the paper is organised as follows. Section 2 reviews the concepts used in the rest of the article. Section 3 presents the feasible split and merge operations. Section 4 presents the learning algorithm and proves its correctness. Section 5 closes the article with some discussion.

# 2 Preliminaries

In this section, we review some concepts from probabilistic graphical models that are used later in this paper. All the graphs and probability distributions in this paper are defined over a finite set of variables $V$. With $|V|$ we mean the number of variables in the $V$ and with $|V_G|$ the number of variables in the graph $G$. Throughout the paper the intended meaning of CGs is multivariate regression CGs if no other interpretation is mentioned. To allow more readable figures bidirected edges are used instead of dashed edges or lines. To not confuse the reader these edges will also be denoted bidirected edges throughout the article.

If a graph $G$ contains an edge between two nodes $V_1$ and $V_2$, we write that $V_1 \to V_2$ is in $G$ for a directed edge, $V_1 \leftrightarrow V_2$ is in $G$ for a bidirected edge, and $V_1 - V_2$ is in $G$ for an undirected edge. With $V_1 \circ\!\!\to V_2$ we mean that either $V_1 \to V_2$ or $V_1 \leftrightarrow V_2$ is in $G$. With $V_1 \circ\!\!-\!\!\circ V_2$ we mean that either $V_1 \to V_2$ or $V_1 - V_2$ is in $G$. With $V_1 \circ\!\!-\!\!\circ V_2$ we mean that there exists an edge between $V_1$ and $V_2$ in $G$. A set of nodes is said to be complete if there exists edges between all pairs of nodes in the set. A complete set of nodes is said to be a clique if there exists no superset of it that is complete.

The parents of a set of nodes $X$ of $G$ is the set $pa_G(X) = \{V_1 | V_1 \to V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The children of $X$ is the set $ch_G(X) = \{V_1 | V_2 \to V_1$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The spouses of $X$ is the set $sp_G(X) = \{V_1 | V_1 \leftrightarrow V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. The adjacents of $X$ is the set $adj_G(X) = \{V_1 | V_1 \to V_2$,$V_1 \leftarrow V_2$, $V_1 \leftrightarrow V_2$ or $V_1 - V_2$ is in $G$, $V_1 \notin X$ and $V_2 \in X\}$. A path from a node $V_1$ to a node $V_n$ in $G$ is a sequence of distinct nodes $V_1, \ldots, V_n$ such that $V_i \in adj_G(V_{i+1})$ for all $1 \leq i < n$. The length of a path is the number of edges in the path. A path is called a cycle if $V_n = V_1$. A path is called descending if $V_i \in pa_G(V_{i+1}) \cup sp_G(V_{i+1})$ for all $1 \leq i < n$. The descendants of a set of nodes $X$ of $G$ is the set $de_G(X) = \{V_n |$ there is a descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. A path is called strictly descending if $V_i \in pa_G(V_{i+1})$ for all $1 \leq i < n$. The strict descendants of a set of nodes $X$ of $G$ is the set $sde_G(X) = \{V_n |$ there is a strict descending path from $V_1$ to $V_n$ in $G$, $V_1 \in X$ and $V_n \notin X\}$. The ancestors (resp. strict ancestors) of $X$ is the set $an_G(X) = \{V_1 | V_n \in de_G(V_1), V_1 \notin X, V_n \in X\}$ (resp. $san_G(X) = \{V_1 | V_n \in sde_G(V_1), V_1 \notin X, V_n \in X\}$). Note that the definition for strict descendants given here coincides to the definition of descendants given by Richardson (2003). Our definition of descendants is however needed for

certain proofs.

A cycle is called a semi-directed cycle if it is descending and $V_i \to V_{i+1}$ is in $G$ for some $1 \leq i < n$. A CG is a graph containing only directed and bidirected edges with no semi-directed cycles. An undirected graph is said to be chordal if every cycle of length four or more has an edge between two non-consecutive vertices in the cycle. A CG is said to be connected if there exists a path between every pair of nodes in it. A connectivity component $C$ of a CG is a maximal set of nodes (wrt to inclusion) st there exists a path between every pair of nodes in $C$ containing only bidirected edges. The connectivity component of a node $X$ in a CG $G$, denoted $co_G(X)$, is the connectivity component in $G$ to which $X$ belongs. A subgraph of $G$ is a subset of nodes and edges in $G$. A subgraph of $G$ induced by a set of its nodes $X$ is the graph over $X$ that has all and only the edges in $G$ whose both ends are in $X$.

A node $C$ is a collider between two nodes $A$ and $B$ in a CG $G$ if there exists edges $A \circ\!\!\to C \leftarrow\!\!\circ B$ in $G$. An unshielded collider is a collider where $A \notin adj_G(B)$ and we then say that $A$ and $B$ have an unshielded collider over $C$. With a non-collider node $C$ between two nodes $A$ and $B$ we mean that $A \circ\!\!-\!\!\circ C \to B$ is in $G$.

Let $X$, $Y$ and $Z$ denote three disjoint subsets of nodes in a CG $G$. $X$ is said to be separated from $Y$ given $Z$ iff there exists no path between any node in $X$ and any node in $Y$ st: (1) every non-collider on the path is not in $Z$ and (2) every collider on the path is in $Z$ or in $san_G(Z)$. We denote this by $X \perp_G Y|Z$. Likewise, we denote by $X \perp_p Y|Z$ that $X$ is independent of $Y$ given $Z$ in a probability distribution $p$. The independence model induced by $G$, denoted as $I(G)$, is the set of separation statements $X \perp_G Y|Z$.

We say that a probability distribution $p$ is Markovian with respect to a CG $G$ when $X \perp_p Y|Z$ if $X \perp_G Y|Z$ for all $X$, $Y$ and $Z$ disjoint subsets of $V$. We say that $p$ is faithful to $G$ when $X \perp_p Y|Z$ iff $X \perp_G Y|Z$ for all $X$, $Y$ and $Z$ disjoint subsets of $V$. We say that two CGs $G$ and $H$ are Markovian equivalent or that they are in the same Markov equivalence class iff $I(G) = I(H)$. If $G$ and $H$ have the same adjacencies and unshielded colliders, then $I(G) = I(H)$ [15, Theorem 1].

# 3 Feasible split and merging

In this section we present the feasible split and feasible merge operations. We also show that there exists a set of CGs with a unique minimal set of bidirected edges for each Markov equivalence class and that these CGs can be reached by repeatedly applying feasible splits to any CG in that Markov equivalence class.

**Lemma 1.** *A CG is in the same Markov equivalence class before and after a feasible split.*

---

**Definition 1.** *Feasible Split*

Let $C$ denote a connectivity component of $G$ and $U$ and $L$ two disjoint subsets of $C$ st $C = U \cup L$ and the subgraph of $G$ induced by $U$ is connected. A split of $C$, performed by replacing every edge $X \leftrightarrow Y$ with $X \rightarrow Y$ st $X \in U$ and $Y \in L$, is feasible iff:

1 For all $A \in sp_G(U) \cap L$, $U \subseteq sp_G(A)$ holds
2 For all $A \in sp_G(U) \cap L$, $pa_G(U) \subseteq pa_G(A)$ holds
3 For all $B \in sp_G(L) \cap U$, $sp_G(B) \cap L$ is a complete set

---

---

**Definition 2.** *Feasible Merging*

Let $U$ and $L$ denote two connectivity components of $G$. A merge between the two components, performed by replacing every edge $X \rightarrow Y$ with $X \leftrightarrow Y$ st $X \in U$ and $Y \in L$, is feasible iff:

1 For all $A \in ch_G(U) \cap L$, $pa_G(U) \cup U \subseteq pa_G(A)$ holds
2 For all $B \in pa_G(L) \cap U$, $ch_G(B) \cap L$ is a complete set
3 $de_G(U) \cap pa_G(L) = \varnothing$

---

*Proof.* Let $G$ be a CG and $G'$ a graph st $G'$ is $G$ with a feasible split performed upon it. $G$ and $G'$ are in different Markov equivalence classes or $G'$ is not a CG iff (1) $G$ and $G'$ do not have the same adjacencies, (2) $G$ and $G'$ do not have the same unshielded colliders or (3) $G'$ contains a semi-directed cycle.

First it is clear that the adjacencies are the same before and after the split since the split does not change the adjacencies in $G$. Secondly let us assume that $G$ and $G'$ do not have the same unshielded colliders. It is clear that a split does not introduce any new unshielded collider, which means that an unshielded collider is removed during the split. Let us say that this unshielded collider is between two nodes $X$ and $Y$ over $Z$ in $G$ st $X$ and/or $Y$ are in $L$ and $Z \in U$. Without loss of generality, let us say that $X$ is in $L$. $X \leftrightarrow Z$ and $Y \circ\!\!\rightarrow Z$ must then hold in $G$ but $X \notin adj_G(Y)$. If $Y \rightarrow Z$ is in $G$ this does not fulfill constraint 2 in definition 1, hence $Y \leftrightarrow Z$ must hold in $G$. Now $Y$ can be either in $U$ or $L$. If $Y \in U$ constraint 1 in definition 1 is violated and if $Y \in L$ constraint 3 in definition 1 is violated. Hence we have a contradiction. Finally let us assume a semi-directed cycle is introduced. This can happen iff we have two nodes $X$ and $Y$ st $X \in de_{G'}(Y)$, $X \in U$ and $Y \in L$. We know no semi-directed cycle existed in $G$ before the split. Then $de_{G'}(Y) \subseteq de_G(Y) \setminus U$ and by definition $X \notin de_G(Y) \setminus U$. Hence we have a contradiction. □

**Lemma 2.** *A CG is in the same Markov equivalence class before and after a feasible merging.*

*Proof.* Let $G$ be a CG and $G'$ a graph st $G'$ is $G$ with a feasible merging performed upon it. $G$ and $G'$ are in different Markov equivalence classes or $G'$ is not a CG iff: (1) $G$ and $G'$ do not have the same adjacencies, (2) $G$ and $G'$ do not have the same unshielded colliders or (3) $G'$ contains a semi-directed cycle.

First it is clear that the adjacencies are the same before and after the merging since the merging does not change the adjacencies in $G$. Secondly let us assume that $G$ and $G'$ do not have the same unshielded colliders. It is clear that a merging does not remove any unshielded colliders which means that there has to exist an unshielded collider between two nodes $X$ and $Y$ over $Z$ in $G'$ that does not exist in $G$. It is clear that the following must hold: $X \notin adj_G(Y)$, $Z \in U$, $X$ and/or $Y \in L$. Without loss of generality, let us say that $X \in L$, which gives us that $X \in ch_G(Z)$. If $Y \in L$ we have $Y \in ch_G(Z)$ which contradicts constraint 2 in definition 2. If $Y \notin L$ we have that $Y \in pa_G(Z)$ or $Y \in sp_G(Z)$, either contradicts constraint 1 in definition 2. Hence an unshielded collider can not be removed. Finally let us assume a semi-directed cycle is introduced. This means that we have three nodes $X, Y$ and $Z$ st $X \in L, Z \in U, Y \notin U \cup L$, $Z \leftrightarrow X \leftarrow Y$ is in $G'$ and $Y \in de_{G'}(Z)$. However, this violates condition 3 in definition 2, because $Y \in pa_G(X)$ and $Y \in de_G(U)$. □

We will now show that there exists a set of CGs which have a unique minimal set of bidirected edges for each Markov equivalence class. We also show that this set of edges is shared by all CGs in the class and that the CGs containing no other bidirected edges than the minimal set, can be reached by repeatedly performing feasible splits on any CG in the class.

**Theorem 1.** *For a Markov equivalence class of CGs, there exists a unique minimal (wrt inclusion) set of bidirected edges that is shared by all members of the class.*

*Proof.* Assume to the contrary that there exists two CGs $G$ and $G'$ st $I(G) = I(G')$ and $G$ and $G'$ have two different minimal sets of bidirected edges. Now for all ordered pair of nodes $A$ and $B$ st $A \leftrightarrow B$ is in $G$ but $A \rightarrow B$ is in $G'$, replace $A \leftrightarrow B$ in $G$ with $A \rightarrow B$ and call this new CG $H$. Obviously $H$ has a proper subset of the bidirected edges in $G$. We can also see that $H$ has no semi-directed cycle, because if it had a semi-directed cycle this cycle would also have been in $G$ or $G'$ which contradicts that $G$ and $G'$ are CGs. Finally we can see that $I(H) = I(G)$ because $H$ has the same adjacencies and unshielded colliders as $G$. To see the last, note that it is impossible that an unshielded collider $C \circ\!\!\rightarrow A \leftrightarrow B$ is in $G$ but not in $H$, because $G'$ has no unshielded collider of $B$ and $C$ over $A$ and $I(G) = I(G')$. □

**Theorem 2.** *A CG has the minimal set of bidirected edges for its Markov equivalence class if no feasible split is possible.*

*Proof.* Assume to the contrary there exists a CG $G$ that does not have the minimal set of bidirected edges for its Markov equivalence class and no split is feasible. Hence there must exist a bidirected edge $X \leftrightarrow Y$ in $G$ st $X \rightarrow Y$ exists in a CG $G'$ st $I(G) = I(G')$. Let $C$ be the component of $X$ in $G$ and $U$ and $L$ be two disjoint subsets of $C$ st $C = U \cup L$, $X \in U$, $Y \in L$ and the subgraph of $G$ induced by $U$ is connected. It is trivial to see that such sets of nodes always must exist. If no split is feasible then we must have that one of the conditions in definition 1 fails for $U$ and $L$. Hence one of the following assumptions must be true.

Assume there exists a node $A$ st $A \in sp_G(U) \cap L$ for which $U \subseteq sp_G(A)$ does not hold. If this is the case there must exist an unshielded collider in $G$ over a node $D$ between $A$ and $E$ st $D \in U, D \in sp_G(A), E \in U, E \in sp_G(D), E \notin sp_G(A)$. This contradicts $I(G) = I(G')$ since $D \rightarrow A$ exists in $G'$.

Assume there exists a node $A$ st $A \in sp_G(U) \cap L$ for which $pa_G(U) \subseteq pa_G(A)$ does not hold. If this is the case there must exist an unshielded collider in $G$ over a node $D$ between $A$ and $E$ st $D \in U, D \in sp_G(A), E \in pa_G(D)$, $E \notin pa_G(A)$. This contradicts $I(G) = I(G')$ since $D \rightarrow A$ exists in $G'$.

Assume there exists a $B \in sp_G(L) \cap U$ for which $sp_G(B) \cap L$ is not complete. If this is the case there must exist an unshielded collider in $G$ over a node $B$ between $D$ and $E$ st $D, E \in L \cap sp_G(B), E \notin adj_G(D)$. This contradicts $I(G) = I(G')$ since $B \rightarrow D$ exists in $G'$.

This shows that if one of the constrains in definition 1 fails we can not have $I(G) = I(G')$ which contradicts the assumption.

□

Finally, it is worth mentioning that we can guarantee that every member of a Markov equivalence class can be reached from any other member of that class by a sequence of feasible splits and mergings. The proof of this result can be seen at *http://www.ida.liu.se/~jospe/pgm12appendix.pdf*. This result is not used in this paper but we conjecture that it will play a central role in future work (see section 6).

# 4 Learning algorithm

In this section we present a constraint based algorithm which learns a CG from a probability distribution faithful to some CG. We then prove that the algorithm is correct and that the returned CG contains exactly the minimal set of bidirected edges for its Markov equivalence class.

The algorithm is very similar to the PC algorithm for directed acyclic graphs [8, 12] and shares the same structure with the learning algorithms presented by both Studený for LWF CGs [13] and Peña for AMP CGs [10]. The algorithm is shown in algorithm 1 and consists of four separate phases. In phase one (line 1-7) the adjacencies of the CG is recovered. In the second phase (line 8) the unshielded colliders are recovered. Phase three (line 9)

then orients some of the remaining edges iff they are oriented in the same direction in all CGs $G'$ st $I(G) = I(G')$. What remains for phase four (line 10-14) is then to orient the rest of the undirected edges st no new unshielded colliders or semi-directed cycles are introduced. This is done according to an algorithm presented in a proof by Koller and Friedman (2009, Theorem 4.13) and is possible since $H_u$ is chordal as shown in Lemma 8.

Learning Algorithm
Given a probability distribution $p$ faithful to an unknown CG $G$, the algorithm learns a CG $H$ st $I(H) = I(G)$. Moreover, $H$ has exactly the minimum set of bidirected edges for its equivalence class.

1 Let $H$ denote the complete undirected graph
2 For $l = 0$ to $l = |V_H| - 2$
3  Repeat while possible
4  Select any ordered pair of nodes $A$ and $B$ in $H$ st $A \in adj_H(B)$ and $|adj(A) \setminus B| \geq l$
5  If there exists $S \subseteq (adj_H(A) \setminus B)$ st $|S| = l$ and $A \perp_p B | S$ then
6  Set $S_{AB} = S_{BA} = S$
7  Remove the edge $A - B$ from $H$
8 Apply rule 0 while possible
9 Apply rules 1-3 while possible
10 Let $H_u$ be the subgraph of $H$ containing only the nodes and the undirected edges in $H$
11 Let $T$ be the clique tree of $H_u$
12 Order the cliques $C_1, ..., C_n$ of $H_u$ st $C_1$ is the root of $T$ and if $C_i$ is closer to the root than $C_j$ in $T$ then $C_i < C_j$.
13 Order the nodes st if $A \in C_i$, $B \in C_j$ and $C_i < C_j$ then $A < B$
14 Orient the undirected edges in $H$ according to the ordering obtained in line 13
15 Return $H$

**Algorithm 1:** Learning algorithm

The rules used in lines 8-9 in algorithm 1 are shown in figure 4.1. A rule is said to be applicable if the antecedent is satisfied for an induced subgraph of $H$. When a rule is applicable one of the non-arrow edge endings is then replaced with an arrow while the rest of the endings are kept the same. Which edge ending is orientated is shown in the consequent of each rule.

A rule is sound if the orientation introduced by the rule must be shared by every CG which contains the antecedent as an induced subgraph.

**Lemma 3.** *The rules 0 - 3 are sound.*

*Proof.* Rule 0: Since $B \notin S_{AC}$, $A \in adj_H(B)$ and $C \in adj_H(B)$ but $A \notin adj_H(C)$ we know that the following configurations can occur: $A \to B \leftarrow C$, $A \leftrightarrow B \leftarrow C$, $A \to B \leftrightarrow C$, $A \leftrightarrow B \leftrightarrow C$. In any other configuration, $B$ would be in every separation set of $A$ and $C$. In all these configurations we have $B \leftarrow\!\circ\, C$, so that edge must exist in $G$. Rule 1: If the edge was not directed

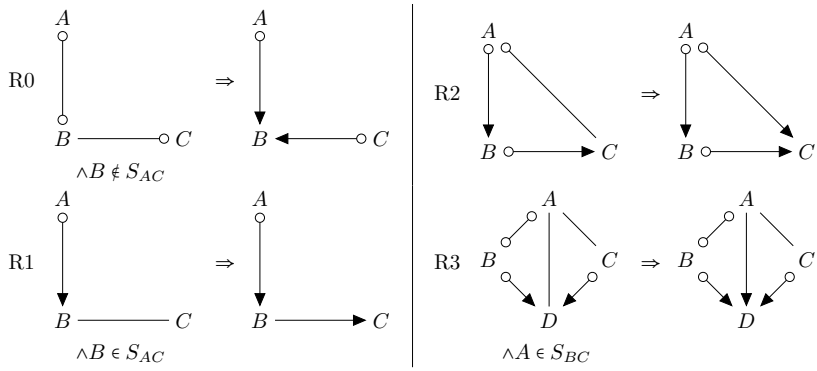



Figure 4.1: The rules

in this direction we would have an unshielded collider of $A$ and $C$ over $B$ that is not in $G$, because $B \in S_{AC}$. Rule 2: If we did not have the edge oriented in this direction we would have a semi-directed cycle in $G$. Rule 3: If the edge was orientated in the opposite direction, by applying rule 2 we would have an unshielded collider of $B$ and $C$ over $A$ that is not in $G$, because $A \in S_{BC}$. □

**Lemma 4.** *$G$ and $H$ have the same adjacencies after line 7.*

*Proof.* Assume to the contrary that there exists an adjacency between two nodes $A$ and $B$ in $G$ st $B \notin de_G(A)$ that does not exist in $H$ or vice versa. We know that, since $G$ is faithful to $p$, all separation statements in $G$ corresponds to independence statements in $p$ and vice versa. We will first cover the case where the adjacency is in $G$ but not in $H$. This means that there exists no separation set $S$ st $A \perp_p B|S$ since $A \perp_G B|S$ does not hold for any $S$. Hence the prerequisite for line 5 in the algorithm can never be true and so the edge between $A$ and $B$ can never be removed and we have a contradiction.

Secondly we will cover if there is an adjacency in $H$ but not in $G$. If there is no adjacency in $G$ between two nodes $A$ and $B$ we know that there has to exist a separation set $S$ st $A \perp_p B|S$ and that $S \subseteq pa_G(A)$ according to the definition of separation. We know, from the paragraph above, that all adjacencies in $G$ must exist in $H$ which means that $pa_G(A) \subseteq adj_H(A)$. However no such set was found in line 5, hence we have a contradiction. □

**Lemma 5.** *$G$ and $H$ have the same unshielded colliders and adjacencies after line 8.*

*Proof.* From Lemma 4 we know that $G$ and $H$ have the same adjacencies. First, assume to the contrary that there exists an unshielded collider in $G$ but not in $H$ after line 8. That means we have an unshielded collider between $A$ and $B$ over $C$ st $C \notin S_{AB}$, $C \in adj_G(A)$ and $C \in adj_G(B)$ but

$A \notin adj_G(B)$. According to Lemma 4 the same adjacencies must also hold in $H$. We know that if we do not have an unshielded collider in $H$ we have $A \circ\!\!-\!\!\circ C \circ\!\!-\!\!\circ B$. This does however fulfill the prerequisite for rule 0, and hence it should have been applied and we have a contradiction.

Secondly it follows from Lemma 3 that rule 0 is sound so there can be no unshielded colliders in $H$ that are not in $G$ after line 8. This brings us to a contradiction. □

**Lemma 6.** *After line 9, $H$ cannot have a subgraph of the form $A \circ\!\!\rightarrow B - C$ without also having the edge $A \rightarrow C$.*

*Proof.* Assume the contrary. First we must have $A \in adj_H(C)$ or rule 1 would orient $B - C$[1]. We can see that $H$ can not have the edge $A \leftarrow\!\!\circ C$ or rule 2 would be applicable for $B - C$. If $H$ has the edge $A \rightarrow C$ we are done, which leaves us with $A - C$.

Secondly we will study why $A$ and $B$ can have an orientation $A \circ\!\!\rightarrow B$. This can be because of one of four reasons.

Case 1: Edge $A \circ\!\!\rightarrow B$ was orientated using rule 0. This means that there exists a node $D$ st $D \circ\!\!\rightarrow B$ and $D \notin adj_H(A)$. $D \in adj_H(C)$ must hold or rule 1 would orient $B - C$[1], but then rule 3 is applicable for the edge $B - C$ so this can not be the reason.

Case 2: Edge $A \circ\!\!\rightarrow B$ was orientated using rule 1. This means that the edge $D \circ\!\!\rightarrow A$ st $D \notin adj(B)$ existed when $A \circ\!\!\rightarrow B$ was oriented. $D \in adj(C)$ must hold or the edge $A - C$ would be oriented by rule 1[1]. Restart the proof with $D \circ\!\!\rightarrow A - C$ and it can be seen that this configuration is impossible. Hence this case can not be the reason.

Case 3: Edge $A \circ\!\!\rightarrow B$ was orientated because of rule 3. This means that we have an unshielded collider of two nodes $D$ and $E$ st $D \circ\!\!\rightarrow B$,$E \circ\!\!\rightarrow B$ $A \in adj_H(D), A \in adj_H(E)$ and $D \notin adj_H(E)$. Now $D \in adj_H(C)$ and $E \in adj_H(C)$ must hold or rule 1 would orient $B - C$[1]. We know that there can be no unshielded collider over $C$ between $D$ and $E$, otherwise rule 3 would be applicable on $A - C$[1]. This gives us that we have $D - C$ and/or $E - C$, since if the edge orientated towards $D$ or $E$ rule 2 would be applicable. However, with this configuration rule 3 is applicable[1] for edge $B - C$ so this can not be the reason.

Case 4: Edge $A \circ\!\!\rightarrow B$ was directed using rule 2. This means that the edges $A \circ\!\!\rightarrow D$ and $D \circ\!\!\rightarrow B$ existed in $H$ when $A \circ\!\!\rightarrow B$ was oriented. $D \in adj_H(C)$, otherwise rule 1 would orient the edge $B - C$[1]. $D - C$ must hold or rule 2 would cause an orientation of $B - C$ or $A - C$. Restart the proof with $A \circ\!\!\rightarrow D - C$ and it can be seen that this configuration is impossible, hence case 4 can not be the reason of the orientation. □

**Lemma 7.** *After line 9, $H$ can not have a subgraph of the form $A \leftrightarrow B - C$.*

[1]Note that if we have $X \circ\!\!-\!\!\circ Y \circ\!\!-\!\!\circ Z$ in $H$ and $X \notin adj_H(Z)$ we must also have $Y \in S_{XY}$ or rule 0 would have been applicable in line 8.

*Proof.* Assume the contrary. Then, according to Lemma 6, $H$ also has the edge $A \to C$. If this is the case then rule 2 is applicable and we have a contradiction. □

**Lemma 8.** *After line 9, removing all directed edges and bidirected edges from $H$ results in a chordal graph.*

*Proof.* Assume to the contrary that there exists a non-chordal cycle $V_1 - V_2 - \ldots - V_n - V_1$ and $n > 3$. We know that there exists a CG $G$ that is faithful to the probability distribution $p$ and has the same unshielded colliders and adjacencies as $H$ by Lemma 5. From Lemma 3 we also know that the rules are sound, i.e. that the orientations in $H$ are in $G$. Hence we know that there still must exist a valid way to orient the edges in $H$ that remain undirected after line 9.

Now if we would orient an edge in the cycle st $V_1 \circ\!\!\to V_2 - \ldots - V_n - V_1$ it is easy to see that we would have to orient every other edge st $V_i \to V_{i+1}$ or we would have a new unshielded collider over some $V_j$ since $V_{j-1} \notin adj_H(V_{j+1})$. If we would orient all edges in this direction we would however have a semi-directed cycle and hence there exists no possible orientation that results in a CG. □

**Lemma 9.** *In line 14, when undirected edges are orientated, no new unshielded colliders are introduced to $H$.*

*Proof.* It follows directly from Lemma 6 that no undirected edge can be orientated such that it creates an unshielded collider with an edge oriented by rules 0-3. Secondly it is proven [5, Theorem 4.13] that no unshilded colliders are created between different undirected edges with the algorithm presented in lines 10-14. □

**Lemma 10.** *No semi-directed cycle exists in $H$ after line 15.*

*Proof.* Assume to the contrary, that a semi-directed cycle exists. If the semi-directed cycle is of length 3 then it could have been be created in one of the following ways:

Case 1: All its edges got oriented by rules 0-3. However, this is a contradiction because rule 2 would be applicable and no semi-directed cycle would remain after applying the rule.

Case 2: All its edges got oriented by line 14. However, this is a contradiction by Theorem 4.13 in Koller and Friedman (2009).

Case 3: Some edges got oriented by rules 0-3 and some by line 14. Then, after applying the rules, the cycle contained the configuration $A \to B - C$ or $A \leftrightarrow B - C$ for some nodes $A, B, C$ in the cycle. The first one is impossible because, otherwise, $A \to C$ by Lemma 6 and, thus, there would not be semi-directed cycle of length 3, which contradicts the paragraph above. The latter one is impossible by Lemma 7.

Hence we can have no semi-directed cycle of length 3. Now assume the semi-directed cycle is of length 4. It could have been be created in one of the following ways:

Case 1: All its edges got oriented by rules 0-3. This implies that the semi-directed cycle in $H$ is actually a cycle with only bidirected edges in $G$ because the rules are sound. However, this implies that there is an unshielded collider in $G$ that was not in $H$, which contradicts Lemma 5, or that $H$ had a semi-directed cycle of length 3, which contradicts the paragraph above.

Case 2: All its edges got oriented by line 14. However, this is a contradiction by Theorem 4.13 in Koller and Friedman (2009).

Case 3: Some edges got oriented by rules 0-3 and some by line 14. Then, after applying the rules, the cycle contained the configuration $A \to B - C$ or $A \leftrightarrow B - C$. The first one is impossible because, otherwise, $A \to C$ by Lemma 6 and there would not be semi-directed cycle. The latter one is impossible by Lemma 7.

Hence $H$ has no semi-directed cycle of length 4. Now, repeating the reasoning for semi-directed cycles of length 5, 6, etc. it is easy to see that no semi-directed cycle can exist since $H$ has a bounded number of nodes. □

**Theorem 3.** *After line 15, $H$ is a CG and $I(H) = I(G)$.*

*Proof.* Lemma 3, 5 and 9 gives that $I(H) = I(G)$ after line 14. Lemma 10 then refutes that $H$ can contain a semi-directed cycle. □

**Theorem 4.** *After line 15, $H$ has exactly the unique minimal set of bidirected edges for its Markov equivalence class.*

*Proof.* It is clear that bidirected edges only can be introduced to $H$ by rules 0-3. From Lemma 3 it also follows that the rules are sound meaning that all the orientations in $H$ caused by the rules have to exist in every $G'$ st $I(G') = I(G)$. □

# 5 Conclusion

In this paper we have presented and proved two fundamental elements for the multivariate regression interpretation of CGs. The first element was an algorithm to learn a multivariate regression CG from a probability distribution faithful to some CG. The second element we have presented is a feasible split and a feasible merging st they alter the structure of a CG but do not change the Markov equivalence class of the CG. We have also shown that there exists a set of CGs with a unique minimal set of bidirected edges for each Markov equivalence class and that these CGs can be reached from any CG in the class using the split operation.

# 6 Further Work

Many elements are still not presented or proven for the multivariate regression interpretation of CGs. The natural continuation of the work presented here would be to develop a learning algorithm with weaker assumptions than the one presented. This could for example be a learning algorithm which only assumes that the probability distribution fullfills the composition property. A second natural continuation of the work here would be to use the split and merging operations to prove that Meek's conjecture holds for the multivariate regression interpretation of CGs, similar way to Peña's work within the LWF interpretation [9]. This could then be used to develop learning algorithms that are correct under the composition property.

# Acknowledgments

This work is funded by the Center for Industrial Information Technology (CENIIT) and a so-called career contract at Linköping University, and by the Swedish Research Council (ref. 2010-4808).

# References

[1] S. A. Andersson, D. Madigan, and M. D. Perlman. An Alternative Markov property for Chain Graphs. *Scandianavian Journal of Statistics*, 28:33–85, 2001.

[2] D. R. Cox and N. Wermuth. Linear Dependencies Represented by Chain Graphs. *Statistical Science*, 8:204–283, 1993.

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

[4] M. Drton. Discrete Chain Graph Models. *Bernoulli*, 15:736–753, 2009.

[5] D. Koller and N. Friedman. *Probabilistic Graphcal Models.* MIT Press, 1999.

[6] S. L. Lauritzen. *Graphical Models.* Clarendon Press, Oxford, 1996.

[7] Z. Ma, X. Xie, and Z. Geng. Structural Learning of Chain Graphs via Decomposition. *Journal of Machine Learning Research*, 9:2847–2880, 2008.

[8] C. Meek. Strong Completeness and Faithfulness in Bayesian networks. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 411–418, 1995.

[9] J. M. Peña. Towards Optimal Learning of Chain Graphs. arXiv:1109.5404v1 [stat.ML], 2011.

[10] J. M. Peña. Learning AMP Chain Graphs under Faithfulness. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, pages 251–258, 2012.

[11] T. S. Richardson. Markov Properties for Acyclic Directed Mixed Graphs. *Scandinavian Journal of Statistics*, 30:145–157, 2003.

[12] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer-Verlag, 1993.

[13] M. Studený. On Recovery Algorithms for Chain Graphs. *International Journal of Approximate Reasoning*, 17:265–293, 1997.

[14] M. Studený, A. Roverato, and Š. Štěpánová. Two Operations of Merging and Splitting Components in a Chain Graph. *Kybernetika*, 45:208–248, 1997.

[15] N. Wermuth and K. Sadeghi. Sequences of Regression and Their Independences. arXiv:1103.2523 [stat.ME], 2012.

## 5.6 Article 5: An inclusion optimal algorithm for chain graph structure learning, with supplement

### Bibliography:

Title: An Inclusion Optimal Algorithm for Chain Graph Structure Learning
Authors: Dag Sonntag, Jose M. Peña and Jens D. Nielsen
Publication: In Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (2014)
Status: Accepted

# An Inclusion Optimal Algorithm for Chain Graph Structure Learning

J. M. Peña [a], Dag Sonntag [a] and Jens D. Nielsen [b]

[a] ADIT, IDA, Linköping University, Sweden
[b] CLC bio, a Quiagen company, Denmark

**Abstract** This paper presents and proves an extension of Meek's conjecture to chain graphs under the Lauritzen-Wermuth-Frydenberg interpretation. The proof of the conjecture leads to the development of a structure learning algorithm that finds an inclusion optimal chain graph for any given probability distribution satisfying the composition property. Finally, the new algorithm is experimentally evaluated.
**Keywords:** Chain Graph, Lauritzen-Wermuth-Frydenberg interpretation, Learning

## 1 Introduction

This paper deals with chain graphs under the Lauritzen-Wermuth-Frydenberg interpretation. Although these chain graphs were introduced to model independencies fairly early [5], there has been relatively little research on them compared to, for example, Bayesian networks. This has mainly to do with the additional complexity that follows from the fact that chain graphs may have both directed and undirected edges, compared to Bayesian networks that only have directed edges. Lately, however, chain graphs have got renewed interest due to their ability to represent more independence models than Bayesian networks [2, 6, 9–15].

A key component that chain graphs still lack compared to Bayesian networks is an inclusion optimal structure learning algorithm. This has mainly to do with that Meek's conjecture has been proven for Bayesian networks [1] but not for chain graphs. We will in this article prove it and, then, use it to develop an algorithm that finds an inclusion optimal chain graph for any probability distribution satisfying the composition property. We will also provide an implementation of the algorithm, and experimentally compare it with the earlier published LCD algorithm [6]. This comparison may seem unfair because the LCD algorithm finds a chain graph a given probability distribution is faithful to and, thus, it imposes a stronger requirement on the input compared to our algorithm. However, this is the best we can do given that we are the first to weaken the faithfulness requirement. To avoid biasing the results in our favour, the experiments only involve faithful probability distributions.

The rest of the article organized as follows. In Section 2, we introduce the notation used in the paper. In Section 3, we prove Meek's conjecture for chain graphs. In section 4, we describe an inclusion optimal learning algorithm. We implement and evaluate the algorithm in Section 5. We close in Section 6 with some conclusions.

# 2 Preliminaries

In this section, we review some concepts from probabilistic graphical models that are used later in this paper. See, for instance, [4] and [12] for further information. All the graphs, independence models, and probability distributions in this paper are defined over a finite set $V$. All the graphs in this paper are hybrid graphs, i.e. they have (possibly) both directed and undirected edges. We assume throughout the paper that the union and the intersection of sets precede the set difference when evaluating an expression.

If a graph $G$ has a directed (resp. undirected) edge between two nodes $X_1$ and $X_2$, then we write that $X_1 \to X_2$ (resp. $X_1 - X_2$) is in $G$. When there is a directed or undirected edge between two nodes of $G$, we say that the two nodes are adjacent in $G$. The parents of a set of nodes $Y$ of $G$ is the set $Pa_G(Y) = \{X_1 | X_1 \to X_2$ is in $G$ and $X_2 \in Y\}$. The neighbors of a set of nodes $Y$ of $G$ is the set $Ne_G(Y) = \{X_1 | X_1 - X_2$ is in $G$ and $X_2 \in Y\}$. The boundary of a node $X_2$ of $G$ is the set $Bd_G(X_2) = Pa_G(X_2) \cup Ne_G(X_2)$. A route between two nodes $X_1$ and $X_n$ of $G$ is a sequence of nodes $X_1, \ldots, X_n$ s.t. $X_i$ and $X_{i+1}$ are adjacent in $G$ for all $1 \leq i < n$. The length of a route is the number of (not necessarily distinct) edges in the route. We treat all singletons as routes of length zero. A route in $G$ is called undirected if $X_i - X_{i+1}$ is in $G$ for all $1 \leq i < n$. A route in $G$ is called descending from $X_1$ to $X_n$ if $X_i - X_{i+1}$ or $X_i \to X_{i+1}$ is in $G$ for all $1 \leq i < n$. If there is a descending route from $X_1$ to $X_n$ in $G$, then $X_n$ is called a descendant of $X_1$. Note that $X_1$ is a descendant of itself, since we allow routes of length zero. The descendants of a set of nodes $Y$ of $G$ is the union of the descendants of each node of $Y$ in $G$ and are denoted $De_G(Y)$. If $X_n$ is a descendant of $X_1$ in $G$ but $X_1$ is not a descendant of $X_n$ in $G$, then $X_n$ is called a strict descendant of $X_1$. The strict descendants of a set of nodes $Y$ of $G$ is the union of the strict descendants of each node of $Y$ in $G$ and are denoted $Sd_G(Y)$. Given a route $\rho$ between $X_1$ and $X_n$ in $G$ and a route $\rho'$ between $X_n$ and $X_m$ in $G$, $\rho \cup \rho'$ denotes the route between $X_1$ and $X_m$ in $G$ resulting from appending $\rho'$ to $\rho$.

A chain is a partition of $V$ into ordered subsets, which we call the blocks of the chain. We say that an element $X \in V$ is to the left of another element $Y \in V$ in a chain $\alpha$ if the block of $\alpha$ containing $X$ precedes the block of $\alpha$ containing $Y$ in $\alpha$. Equivalently, we can say that $Y$ is to the right of $X$ in $\alpha$. We say that a graph $G$ and a chain $\alpha$ are consistent when (i) for every edge $X \to Y$ in $G$, $X$ is to the left of $Y$ in $\alpha$, and (ii) for every edge $X - Y$ in $G$, $X$ and $Y$ are in the same block of $\alpha$. A chain graph (CG) is a graph

that is consistent with a chain. A set of nodes of a CG is connected if there exists an undirected route in the CG between every pair of nodes of the set. A component of a CG is a maximal (wrt set inclusion) connected set of its nodes. A block of a CG is a set of components of the CG s.t. there is no directed edge between their nodes in the CG. Note that a component of a CG is connected, whereas a block of a CG or a block of a chain that is consistent with a CG is not necessarily connected. Given a set $K$ of components of $G$, a component $C \in K$ is called maximal in $G$ if none of its nodes is a descendant of $K \setminus \{C\}$ in $G$. A component $C$ of $G$ is called terminal in $G$ if its descendants in $G$ are exactly $C$. Let a component $C$ of $G$ be partitioned into two non-empty connected subsets $C \setminus L$ and $L$. By splitting $C$ into $C \setminus L$ and $L$ in $G$, we mean replacing every edge $X - Y$ in $G$ s.t. $X \in C \setminus L$ and $Y \in L$ with an edge $X \to Y$. Moreover, we say that the split is feasible if (i) $X - Y$ is in $G$ for all $X, Y \in Ne_G(L) \cap (C \setminus L)$, and (ii) $X \to Y$ is in $G$ for all $X \in Pa_G(L)$ and $Y \in Ne_G(L) \cap (C \setminus L)$. Let $L$ and $R$ denote two components of $G$ s.t. $Pa_G(R) \cap L \neq \emptyset$. By merging $L$ and $R$ in $G$, we mean replacing every edge $X \to Y$ in $G$ s.t. $X \in L$ and $Y \in R$ with an edge $X - Y$. Moreover, we say that the merging is feasible if (i) $X - Y$ is in $G$ for all $X, Y \in Pa_G(R) \cap L$, and (ii) $X \to Y$ is in $G$ for all $X \in Pa_G(R) \setminus L$ and $Y \in Pa_G(R) \cap L$.

A section of a route $\rho$ in a CG is a maximal undirected subroute of $\rho$. A section $X_2 - \ldots - X_{n-1}$ of $\rho$ is a collider section of $\rho$ if $X_1 \to X_2 - \ldots - X_{n-1} \leftarrow X_n$ is a subroute of $\rho$. Moreover, the edges $X_1 \to X_2$ and $X_{n-1} \leftarrow X_n$ are called collider edges. Let $X$, $Y$ and $Z$ denote three disjoint subsets of $V$. A route $\rho$ in a CG is said to be $Z$-active when (i) every collider section of $\rho$ has a node in $Z$, and (ii) every non-collider section of $\rho$ has no node in $Z$. When there is no route in a CG $G$ between a node of $X$ and a node of $Y$ that is $Z$-active, we say that $X$ is separated from $Y$ given $Z$ in $G$ and denote it as $X \perp_G Y|Z$. We denote by $X \not\perp_G Y|Z$ that $X \perp_G Y|Z$ does not hold.

Let $X$, $Y$, $Z$ and $W$ denote four disjoint subsets of $V$. An independence model $M$ is a set of statements of the form $X \perp_M Y|Z$, meaning that $X$ is independent of $Y$ given $Z$. Given two independence models $M$ and $N$, we denote by $M \subseteq N$ that if $X \perp_M Y|Z$ then $X \perp_N Y|Z$. We say that $M$ is a graphoid if it satisfies the following properties: Symmetry $X \perp_M Y|Z \Rightarrow Y \perp_M X|Z$, decomposition $X \perp_M Y \cup W|Z \Rightarrow X \perp_M Y|Z$, weak union $X \perp_M Y \cup W|Z \Rightarrow X \perp_M Y|Z \cup W$, contraction $X \perp_M Y|Z \cup W \wedge X \perp_M W|Z \Rightarrow X \perp_M Y \cup W|Z$, and intersection $X \perp_M Y|Z \cup W \wedge X \perp_M W|Z \cup Y \Rightarrow X \perp_M Y \cup W|Z$. An independence model $M$ is also said to satisfy the composition property when $X \perp_M Y|Z \wedge X \perp_M W|Z \Rightarrow X \perp_M YW|Z$. The independence model induced by a CG $G$, denoted as $I(G)$, is the set of separation statements $X \perp_G Y|Z$ that holds in $G$. It is known that $I(G)$ is a graphoid [13, Lemma 3.1]. Let $H$ denote the graph resulting from a feasible split or merging in a CG $G$. Then, $H$ is a CG and $I(H) = I(G)$ [14, Lemma 5 and Corollary 9]. Two CGs $H$ and $G$ are said to be in the same

equivalence class if $I(H) = I(G)$.

A CG $G$ is an independence (I) map of an independence model $M$ if $I(G) \subseteq M$. Moreover, $G$ is a minimal independence (MI) map of $M$ if removing any edge from $G$ makes it cease to be an I map of $M$. An independence model $M$ is said to be faithful w.r.t. a CG $G$ if $I(G) = M$. We also say that an independence model $M$ is faithful if it is faithful to some CG $G$. A CG $G$ is said to satisfy the local Markov property w.r.t. an independence model $M$ iff $X \perp_M V \setminus X \setminus Sd_G(X) \setminus Bd_G(X) | Bd_G(X)$ for all $X \in V$. Given any chain $C_1, \ldots, C_n$ that is consistent with $G$, we say that $G$ satisfies the pairwise block-recursive Markov property w.r.t. $M$ if $X \perp_M Y | \cup_{j=1}^{k^*} C_j \setminus \{X, Y\}$ for all non-adjacent nodes $X$ and $Y$ of $G$ and where $k^*$ is the smallest $k$ s.t. $X, Y \in \cup_{j=1}^{k} C_j$. If $M$ is a graphoid and $G$ satisfies the local Markov property or the pairwise block-recursive Markov property w.r.t. $M$, then $G$ is an I map of $M$ [4, Theorem 3.34]. We say that a CG $G_\alpha$ is a MI map of an independence model $M$ relative to a chain $\alpha$ if $G_\alpha$ is a MI map of $M$ and $G_\alpha$ is consistent with $\alpha$. A CG $G$ is said to include an independence model $M$ iff $I(G) \subseteq M$. $G$ is also said to be inclusion optimal w.r.t. $M$ iff $I(G) \subseteq M$ and there exists no other CG $H$ such that $I(G) \subset I(H) \subseteq M$.

# 3 Meek's conjecture for chain graphs

In this section we extend Meek's conjecture to chain graphs. This will later be used to prove that the CG structure learning algorithm provided in the next section is inclusion optimal for those probability distributions that satisfy the composition property.

Given two directed and acyclic graphs $G$ and $H$ s.t. $I(H) \subseteq I(G)$, Meek's conjecture states that we can transform $G$ into $H$ by a sequence of directed edge additions and covered edge reversals s.t. after each operation in the sequence $G$ is a directed and acyclic graph and $I(H) \subseteq I(G)$ [7]. Meek's conjecture was proven to be true in [1, Theorem 4] by developing an algorithm that constructs a valid sequence of operations. In this section, we extend Meek's conjecture from directed and acyclic graphs to CGs, and prove that the extended conjecture is true. Specifically, given two CGs $G$ and $H$ s.t. $I(H) \subseteq I(G)$, we prove that $G$ can be transformed into $H$ by a sequence of directed and undirected edge additions and feasible splits and mergings s.t. after each operation in the sequence $G$ is a CG and $I(H) \subseteq I(G)$. The proof is constructive in the sense that we give an algorithm that constructs a valid sequence of operations. See Appendix A for an example run of the algorithm.

We start by introducing two new operations on CGs. It is worth mentioning that all the algorithms in this paper use a "by reference" calling convention, meaning that the algorithms can modify the arguments passed to them. Let $K$ denote a block of a CG $G$. Let $L \subseteq K$. By feasible block splitting (fbsplitting) $K$ into $K \setminus L$ and $L$ in $G$, we mean running the algo-

Fbsplit($K$, $L$, $G$)

1 Let $L_1, \ldots, L_n$ denote the maximal connected subsets of $L$ in $G$
2 For $i = 1$ to $n$ do
3  Add an edge $X - Y$ to $G$ for all $X, Y \in Ne_G(L_i) \cap (K \setminus L)$
4  Add an edge $X \to Y$ to $G$ for all $X \in Pa_G(L_i)$ and $Y \in Ne_G(L_i) \cap (K \setminus L)$
5 For $i = 1$ to $n$ do
6  Let $K_j$ denote the component of $G$ s.t. $L_i \subseteq K_j$
7  If $K_j \setminus L_i \neq \emptyset$ then
8  Split $K_j$ into $K_j \setminus L_i$ and $L_i$ in $G$

Fbmerge($L$, $R$, $G$)

1 Let $R_1, \ldots, R_n$ denote the components of $G$ that are in $R$
2 For $i = 1$ to $n$ do
3  Add an edge $X - Y$ to $G$ for all $X, Y \in Pa_G(R_i) \cap L$
4  Add an edge $X \to Y$ to $G$ for all $X \in Pa_G(R_i) \setminus L$ and $Y \in Pa_G(R_i) \cap L$
5 For $i = 1$ to $n$ do
6  Let $L_j$ denote the component of $G$ st $L_j \subseteq L \cup R$ and $Pa_G(R_i) \cap L_j \neq \emptyset$
7  If $L_j \neq \emptyset$ then
8  Merge $L_j$ and $R_i$ in $G$

Figure 1: Fbsplit And Fbmerge.

rithm at the top of Figure 1. The algorithm repeatedly splits a component of $G$ until $L$ becomes a block of $G$. Before the splits, the algorithm adds to $G$ the smallest set of edges so that the splits are feasible. Let $L$ and $R$ denote two blocks of a CG $G$. By feasible block merging (fbmerging) $L$ and $R$ in $G$, we mean running the algorithm at the bottom of Figure 1. The algorithm repeatedly merges two components of $G$ until $L \cup R$ becomes a block of $G$. Before the mergings, the algorithm adds to $G$ the smallest set of edges so that the mergings are feasible. It is worth mentioning that the component $L_j$ in line 6 is guaranteed to be unique by the edges added in lines 3 and 4.

Our proof of the extension of Meek's conjecture to CGs builds upon an algorithm for efficiently deriving the MI map $G_\alpha$ of the independence model induced by a given CG $G$ relative to a given chain $\alpha$. The pseudocode of the algorithm, called Method B3, can be seen in Figure 2. Method B3 works

Construct $\beta(G, \alpha, \beta)$

1 Set $\beta = \varnothing$
2 Set $H = G$
3 Let $C$ denote any terminal component of $H$ whose leftmost node in $\alpha$ is rightmost in $\alpha$
4 Add $C$ as the leftmost block of $\beta$
5 Let $R$ denote the right neighbor of $C$ in $\beta$
6 If $R \neq \varnothing$, $Pa_G(R) \cap C = \varnothing$, and the nodes of $C$ are to the right of the nodes of $R$ in $\alpha$ then
7 Replace $C, R$ with $R, C$ in $\beta$
8 Go to line 5
9 Remove $C$ and all its incoming edges from $H$
10 If $H \neq \varnothing$ then
11 Go to line 3

Method B3($G$, $\alpha$)

1 Construct $\beta(G, \alpha, \beta)$
2 Let $C$ denote the rightmost block of $\alpha$ that has not been considered before
3 Let $K$ be the leftmost block of $\beta$ s.t. $K \cap C \neq \varnothing$
4 Set $L = K \cap C$
5 If $K \setminus L \neq \varnothing$ then
6 Fbsplit($K$, $L$, $G$)
7 Replace $K$ with $K \setminus L, L$ in $\beta$
8 Let $R$ denote the right neighbor of $L$ in $\beta$
9 If $R \neq \varnothing$ and some node of $R$ is not to the right of the nodes of $L$ in $\alpha$
10 Fbmerge($L$, $R$, $G$)
11 Replace $L, R$ with $L \cup R$ in $\beta$
12 Go to line 3
13 If $\beta \neq \alpha$ then
14 Go to line 2

Figure 2: Method B3.

iteratively by fbsplitting and fbmerging some blocks of $G$ until the resulting CG is consistent with $\alpha$. It is not difficult to see that such a way of working results in a CG that is an I map of $I(G)$. However, in order to arrive at $G_\alpha$, the blocks of $G$ to modify in each iteration must be carefully chosen. For this purpose, Method B3 starts by calling Construct $\beta$ to derive a chain $\beta$ that is consistent with $G$ and as close to $\alpha$ as possible (see lines 5-8). By $\beta$ being as close to $\alpha$ as possible, we mean that the number of blocks Method B3 will later fbsplit and fbmerge is kept at a minimum, because Method B3 will use $\beta$ to choose the blocks to modify in each iteration. A line of Construct $\beta$ that is worth explaining is line 3, because it is crucial for the correctness of Method B3 (see Case 3.2.4 in the proof of Lemma 2). This line determines the order in which the components of $H$ (initially $H = G$) are added to $\beta$ (initially $\beta = \varnothing$). In principle, a component of $H$ may have nodes from several blocks of $\alpha$. Line 3 labels each terminal component of $H$ with its leftmost node in $\alpha$ and, then, chooses any terminal component whose label node is rightmost in $\alpha$. This is the next component to add to $\beta$.

Once $\beta$ has been constructed, Method B3 proceeds to transform $G$ into $G_\alpha$. In particular, Method B3 considers the blocks of $\alpha$ one by one in the reverse order in which they appear in $\alpha$. For each block $C$ of $\alpha$, Method B3 iterates through the following steps. First, it finds the leftmost block $K$ of $\beta$ that has some nodes from $C$. These nodes, denoted as $L$, are then moved to the right in $\beta$ by fbsplitting $K$ to create a new block $L$ of $G$ and $\beta$. If the nodes of the right neighbor $R$ of $L$ in $\beta$ are to the right of the nodes of $L$ in $\alpha$, then Method B3 is done with $C$. Otherwise, Method B3 moves $L$ further to the right in $\beta$ by fbmerging $L$ and $R$ in $G$ and $\beta$.

The proof of the following important intermediate result can be found in Appendix B.

**Lemma 1.** *Let $M$ denote an independence model, and $\alpha$ a chain $C_1, \ldots, C_n$. If $M$ is a graphoid, then there exits a unique CG $G_\alpha$ that is a MI map of $M$ relative to $\alpha$. Specifically, for each node $X$ of each block $C_k$ of $\alpha$, $Bd_{G_\alpha}(X)$ is the smallest subset $B$ of $\cup_{j=1}^{k} C_j \setminus \{X\}$ s.t. $X \perp_M \cup_{j=1}^{k} C_j \setminus \{X\} \setminus B | B$.*[1]

The proof of the following lemma, which guarantees the correctness of Method B3, can be found in Appendix B.

**Lemma 2.** *Let $G_\alpha$ denote the MI map of the independence model induced by a CG $G$ relative to a chain $\alpha$. Then, Method B3($G$, $\alpha$) returns $G_\alpha$.*

We are now ready to prove that the extension of Meek's conjecture to CGs is true. The proof is constructive in the sense that we give an algorithm that constructs a valid sequence of operations. The pseudocode of our algorithm, called Method G2H, can be seen in Figure 3. The proof of the following theorem, which guarantees the correctness of Method G2H, can be found in Appendix B.

[1] By convention, $X \perp_M \varnothing | \cup_{j=1}^{k} C_j \setminus \{X\}$.

Method G2H($G$, $H$)

1 Let $\alpha$ denote a chain that is consistent with $H$
2 Method B3($G$, $\alpha$)
3 Add to $G$ the edges that are in $H$ but not in $G$

Figure 3: Method G2H.

**Theorem 1.** *Given two CGs $G$ and $H$ s.t. $I(H) \subseteq I(G)$, Method G2H($G$, $H$) transforms $G$ into $H$ by a sequence of directed and undirected edge additions and feasible splits and mergings s.t. after each operation in the sequence $G$ is a CG and $I(H) \subseteq I(G)$.*

# 4 The CKES algorithm

With the extension of Meek's conjecture proven, we can now present an algorithm which relies upon it to find an inclusion optimal CG structure for a given probability distribution that satisfies the composition property. The algorithm is based on random walks in the equivalence classes and is a generalized version of the KES algorithm used for Bayesian network structure learning [8]. Unlike KES, the CKES algorithm is not score-based but constraint-based. The algorithm is shown in Figure 4. The proof of the theorem below, which guarantees the correctness of the CKES algorithm, can be found in Appendix B.

**Theorem 2.** *For any probability distribution $p$ for which the composition property holds, the CKES algorithm finds a CG that is inclusion optimal w.r.t. $p$.*

It is worth mentioning that it follows from the proof of the theorem above that the four operations in the CKES algorithm are necessary to guarantee its correctness.

Due to the infeasibility to enumerate all CGs in each visited equivalence class and the unavailability of the probability distribution $p$, some approximations had to be made when implementing the CKES algorithm. There are two major differences between the algorithm described in Figure 4 and its implementation in Figure 5. The first is that we check whether an independence holds in $p$ by running a hypothesis test on a sample from it, which is what we have access to in practice. Such a test is represented in Figure 5 with the function $I(X, Y|Z)$, which returns the $p$-value of the test. The second major difference is that line 2 in Figure 4 has been replaced by two loops visible in lines 2 and 4 in Figure 5. The loop in line 4 considers $m$ graphs in the current equivalence class to find the best edge addition or removal for that class. However, unless $m \to \infty$ it is unclear whether all graphs

CKES algorithm

1 $G$ =Empty graph
2 Repeat until all the CGs in the equivalence class of $G$ have been considered:
3 For every ordered pair of nodes $X$ and $Y$:
4 If $X \to Y$ is in $G$ but $X \perp_p Y | Bd_G(Y) \setminus X$ then remove $X \to Y$ from $G$ and go to line 2
5 If $X - Y$ is in $G$ but $X \perp_p Y | Bd_G(Y) \setminus X$ and $X \perp_p Y | Bd_G(X) \setminus Y$ then remove $X - Y$ from $G$ and go to line 2
6 If $X \to Y$ is not in $G$ but adding $X \to Y$ to $G$ results in a CG and $X \not\perp_p Y | Bd_G(Y)$ then add $X \to Y$ to $G$ and go to line 2
7 If $X - Y$ is not in $G$ but adding $X - Y$ to $G$ results in a CG and $X \not\perp_p Y | Bd_G(Y)$ or $X \not\perp_p Y | Bd_G(X)$ then add $X - Y$ to $G$ and go to line 2
8 Move to another CG in the same equivalence class of $G$ by performing a random number of random feasible merges or feasible splits on $G$ and thereby updating $G$
9 Return G

Figure 4: The CKES Algorithm.

in the equivalence class have been considered. This means that not all edge additions and removals may have been considered for the equivalence class and that the edge addition or edge removal with lowest resp. highest $p$-value may not have been found. This can be acceptable if some edge addition or removal is found but not when determining if the equivalence class is an inclusion optimal CG (i.e. when terminating the algorithm). The loop in line 2 has therefore been added to make sure that the final equivalence class is searched extra thoroughly by repeating the inner loop $k$ times. Finally, there is also the $l$ variable in line 10, which controls how many splits or mergings that are performed when moving between graphs in the same equivalence class. All in all this means that large values on $k$, $l$ and $m$ increase the probability that the implementation terminates in an inclusion optimal CG, assuming that the independence hypothesis test does not induce any error. Theoretically, an inclusion optimal CG can only be guaranteed to be found if either $m \to \infty$ or $k \to \infty$. Empirical results do however suggest that these conditions can be relaxed considerably and that the parameter values $k = 10$, $l = 4$, $m = 100$ are high enough to find a local optimum for our experiments in the next section.

CKES implementation$(\alpha, k, l, m)$

1 $G$ =Empty graph
2 Repeat until $I(G)$ does not change for $k$ iterations:
3 $p_r = 0$, $p_a = 1$
4 Repeat for $m$ iterations:
5 For every ordered pair of nodes $X$ and $Y$:
6 If $X \to Y$ is in $G$ but $I(X, Y|Bd_G(Y) \setminus X) > p_r$, then set $p_r = I(X, Y|Bd_G(Y) \setminus X)$ and $G_r = G \setminus \{X \to Y\}$
7 If $X - Y$ is in $G$ but $min(I(X, Y|Bd_G(Y) \setminus X), I(X, Y|Bd_G(X) \setminus Y)) > p_r$, then set $p_r = min(I(X, Y|Bd_G(Y) \setminus X), I(X, Y|Bd_G(X) \setminus Y))$ and $G_r = G \setminus \{X - Y\}$
8 If $X \to Y$ is not in $G$ but adding $X \to Y$ to $G$ results in a CG and $I(X, Y|Bd_G(Y)) < p_a$, then set $p_a = I(X, Y|Bd_G(Y))$ and $G_a = G \cup \{X \to Y\}$
9 If $X - Y$ is not in $G$ but adding $X - Y$ to $G$ results in a CG and $min(I(X, Y|Bd_G(Y)), I(X, Y|Bd_G(X)) < p_a$, then set $p_a = min(I(X, Y|Bd_G(Y)), I(X, Y|Bd_G(X))$ and $G_a = G \cup \{X - Y\}$
10 Move to another CG in the same equivalence class of $G$ by performing $l$ random feasible merges or feasible splits on $G$ and thereby updating $G$
11 If $p_r > \alpha$, then set $G = G_r$ and go to line 2
12 If $p_a \leq \alpha$, then set $G = G_a$ and go to line 2
13 Return $G$

Figure 5: CKES Implementation.

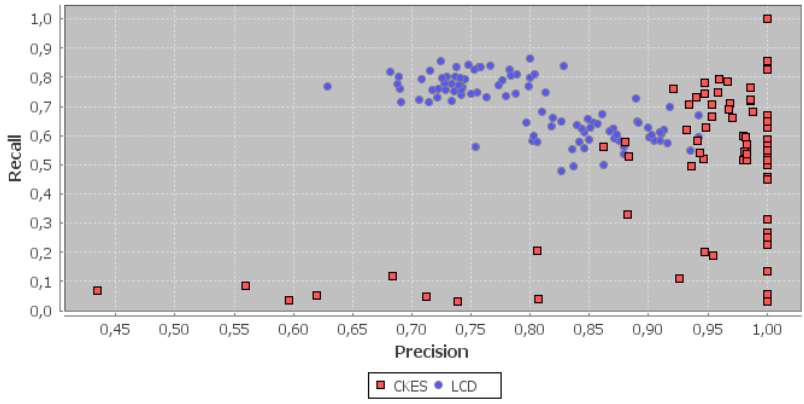



Figure 6: Multiple Locally Optimal CGs May Exist.

# 5 Evaluation

To our knowledge, only two other CG structure learning algorithms have been presented under the Lauritzen-Wermuth-Frydenberg interpretation. These are the LCG recovery algorithm [11] and the LCD algorithm [6]. The SINful approach [3] can also be used to learn a CG structure but only if its components and their order is known in advance. Both the LCG recovery algorithm and the LCD algorithm assume faithfulness. It can be shown that if this assumption does not hold, then these algorithms may find a CG that does not include the probability distribution at hand, even if the algorithms are given perfect information about the independencies that hold in the probability distribution (see Appendix C). In this section we will evaluate our algorithm only against the LCD algorithm, because it is the only that has been presented with an implementation and evaluation.

Recall that for the CKES algorithm to find an inclusion optimal CG of a probability distribution, it suffices that the latter satisfies the composition property. However, the LCD algorithm finds an inclusion optimal CG of a probability distribution only if the latter satisfies the faithfulness assumption. To avoid biasing the experimental results in our favour, we only considered faithful probability distributions in the experiments (see Appendix C for experiments under the composition property assumption). Specifically, we considered CGs with P=10, 20 nodes and N=2, 5 adjacencies per node on average. For each of these four scenarios, we generated 10 random CG structures with the algorithm described in [6]. Each structure was then parameterized into one binary and one Gaussian probability distribution as proposed by [6]. Then, samples of size n=100, 30000 were obtained from these probability distributions. Using these samples and the independence hypothesis tests provided by Ma et al. with significance level

$\alpha$ = 0.01, the CKES and LCD algorithms were finally run. Note that the probability distributions sampled are likely to satisfy the faithfulness assumption, but there is no guarantee. Note also that the samples can have additional independencies that are not in the sampled probability distributions. As performance measures in the experiments, we computed the recall of the separations in the CGs sampled, and the precision of the separations in the CGs learnt. Since time did not permit it, we did not check all the separations but a large number of uniformly selected separations. Specifically, for any given CG, we uniformly selected 10000 triplets $(X, Y, Z)$ with $X$, $Y$ and $Z$ disjoint sets of nodes among all possible such triplets. For each triplet $(X, Y, Z)$, we tested whether $X$ was separated from $Y$ given $Z$ in the given CG. If so, the separation was selected.

The first conclusion that we drew from our experiments is that multiple locally optimal CGs may be found from the same sample. This phenomenon has been previously noted for Bayesian networks [8]. Figure 6 illustrates this phenomenon. Specifically, it plots the recall and precision of CGs obtained by running 100 times the LCD and 100 times the CKES algorithm on the same 30000 samples from a Gaussian probability distribution sampled from a CG with P=10 and N=2. In the case of the LCD implementation this randomization comes from variations in the order in which the variables are considered by the algorithm. The CKES implementation on the other hand contains both the explicit randomness due to lines 4 and 10 in Figure 5 and also the implicit randomness due to the order in which the variables are chosen in line 5 in Figure 5. The explicit randomness can to a large extent be minimized with large enough values on the parameters $l$ and $m$ in the algorithm. Increasing these parameters significantly was however seen to have little impact on the results. Instead, it is the implicit randomness that is most significant.

With the numerous locally optimal CGs found in mind, we then set out to find which of the two algorithms reached the best CG. This was done by running both algorithms 100 times on each sample set in the experiments and, then, selecting the best CGs in terms of recall and precision from those obtained from each sample set. Table 1 presents average recall and precision values over the best CGs obtained from the 10 sample sets created for each scenario in the experiments. In addition to these results, Table 2 presents the following values for each scenario:

- Best recall: The number of sample sets on which the CKES (resp. LCD) algorithm learnt a CG with better or equal recall than any of the CGs learnt by the LCD (resp. CKES) algorithm.
- Best precision: The number of sample sets on which the CKES (resp. LCD) algorithm learnt a CG with better or equal precision than any of the CGs learnt by the LCD (resp. CKES) algorithm.
- Best recall and precision: The number of sample sets on which the CKES (resp. LCD) algorithm learned a CG with better or equal recall

Table 1: Average Results For Categorical Data (Top) And Gaussian Data (Bottom).

| CG structure | Recall | | Precision | |
|---|---|---|---|---|
| | CKES | LCD | CKES | LCD |
| P=10,N=2,n=100 | 0.97 | 0.98 | 0.60 | 0.59 |
| P=10,N=2,n=30000 | 0.96 | 0.96 | 0.95 | 0.87 |
| P=10,N=5,n=100 | 1.00 | 1.00 | 0.07 | 0.06 |
| P=10,N=5,n=30000 | 0.95 | 0.98 | 0.92 | 0.34 |
| P=20,N=2,n=100 | 0.97 | 0.99 | 0.65 | 0.64 |
| P=20,N=2,n=30000 | 0.89 | 0.95 | 0.92 | 0.83 |
| P=20,N=5,n=100 | 0.98 | 1.00 | 0.17 | 0.15 |
| P=20,N=5,n=30000 | 0.71 | 0.95 | 0.89 | 0.56 |

| CG structure | Recall | | Precision | |
|---|---|---|---|---|
| | CKES | LCD | CKES | LCD |
| P=10,N=2,n=100 | 0.95 | 0.99 | 0.83 | 0.76 |
| P=10,N=2,n=30000 | 0.93 | 0.99 | 1.00 | 0.99 |
| P=10,N=5,n=100 | 0.91 | 0.96 | 0.44 | 0.11 |
| P=10,N=5,n=30000 | 1.00 | 0.92 | 1.00 | 0.57 |
| P=20,N=2,n=100 | 0.92 | 0.99 | 0.99 | 0.87 |
| P=20,N=2,n=30000 | 0.94 | 1.00 | 1.00 | 0.98 |
| P=20,N=5,n=100 | 0.65 | 1.00 | 0.72 | 0.13 |
| P=20,N=5,n=30000 | 0.77 | 0.91 | 1.00 | 0.53 |

and precision than any of the CGs learnt by the LCD (resp. CKES) algorithm.

We can now make the following observations about the results obtained:

1. According to Table 1, none of the algorithms is generally able to recreate the CG sampled since their recall and precision values do not equal 1. This means that the sets of independencies detected by the hypothesis test on the sample sets are not faithful to the CGs that generated the sample sets. As we will see, this leads the two algorithms in the evaluation to behave differently.
2. According to Table 1, the LCD algorithm on average achieves good recall but not so good precision. The reason for this may be that the faithfulness assumption makes the LCD algorithm search for a CG that represents all the independencies that are detected in the sample set. However, such a CG may also represent many other independencies. Therefore, the LCD algorithm trades precision for recall.
3. According to Table 1, the CKES algorithm on average achieves good precision but not so good recall. The reason for this may be that the composition property assumption makes the CKES algorithm search

Table 2: Best Results For Categorical Data (Top) And Gaussian Data (Bottom).

| CG structure | Best recall | | Best precision | | Best recall and precision | |
|---|---|---|---|---|---|---|
| | CKES | LCD | CKES | LCD | CKES | LCD |
| P=10,N=2,n=100 | 6 | 9 | 6 | 4 | 3 | 3 |
| P=10,N=2,n=30000 | 8 | 5 | 9 | 1 | 8 | 1 |
| P=10,N=5,n=100 | 10 | 10 | 9 | 1 | 6 | 1 |
| P=10,N=5,n=30000 | 6 | 10 | 10 | 0 | 6 | 0 |
| P=20,N=2,n=100 | 2 | 9 | 8 | 2 | 0 | 2 |
| P=20,N=2,n=30000 | 1 | 9 | 10 | 0 | 1 | 0 |
| P=20,N=5,n=100 | 3 | 10 | 9 | 1 | 3 | 1 |
| P=20,N=5,n=30000 | 0 | 10 | 10 | 0 | 0 | 0 |

| CG structure | Best recall | | Best precision | | Best recall and precision | |
|---|---|---|---|---|---|---|
| | CKES | LCD | CKES | LCD | CKES | LCD |
| P=10,N=2,n=100 | 3 | 9 | 9 | 2 | 1 | 2 |
| P=10,N=2,n=30000 | 3 | 10 | 10 | 9 | 3 | 9 |
| P=10,N=5,n=100 | 6 | 9 | 10 | 0 | 6 | 0 |
| P=10,N=5,n=30000 | 9 | 2 | 10 | 0 | 8 | 0 |
| P=20,N=2,n=100 | 1 | 9 | 10 | 0 | 1 | 0 |
| P=20,N=2,n=30000 | 2 | 9 | 9 | 7 | 2 | 7 |
| P=20,N=5,n=100 | 1 | 10 | 10 | 0 | 1 | 0 |
| P=20,N=5,n=30000 | 4 | 6 | 10 | 0 | 4 | 0 |

for a CG that only represent independencies that are detected in the sample set. However, such a CG may not represented many of the detected independencies. Therefore, the CKES algorithm trades off recall for precision. This can also be seen in Figure 6 where the CKES algorithm achieves good recall only if it also achieves good precision, which indicates that it considers precision as more important.

In other words, it seems that the faithfulness assumption makes the LCD algorithm overconfident and aggressive, whereas the composition property assumption makes the CKES algorithm cautious and conservative. As we have seen, this implies that in practice the LCD algorithm prioritizes recall and the CKES algorithm precision. In our opinion, the latter behaviour is to be preferred: By achieving better precision, the CGs learnt by the CKES algorithm represent to a greater extent an I map of the probability distribution sampled. And this is the goal in structural learning.

4. The last observation is about which algorithm usually returns the best CG. We can see in Table 2 that the LCD algorithm usually returns the best CG in terms of recall, whereas the CKES algorithm usually returns the best CG in terms of precision. However, note that it is the CKES algorithm which usually returns the best CG in terms of recall and precision jointly.

# 6 Conclusions

In this article, we have presented and proved an extension of Meek's conjecture to CGs under the Lauritzen-Wermuth-Frydenberg interpretation. We have also shown how this conjecture can be used to develop a CG structure learning algorithm that is inclusion optimal for those probability distributions that satisfy the composition property. This property is weaker than faithfulness, which has been a prerequisite for all earlier presented algorithms. We think that the extension of Meek's conjecture proven in this paper is very relevant, since we believe that it will be used by other researchers to propose their own inclusion optimal learning algorithms under the composition property assumption.

In our experiments, we have seen that the CKES algorithm, which only assumes the composition property, produces CGs that can be interpreted as I maps of the probability distribution at hand to a greater extent than the CGs produced by the LCD algorithm, which assumes faithfulness. The reason seems to be that assuming only the composition property induces a more conservative behaviour as compared to assuming faithfulness. CGs are aimed at helping discovering independencies and/or performing probabilistic inference faster but accurately. In both of these tasks it is more important that the independencies in the learnt CG are true in the probability distribution at hand than that all the true independencies are in the CG. In other words, precision is more important than recall for these tasks. The CKES algorithm is therefore, in our humble opinion, preferred to the other algorithms available today.

In our experiments, we have also seen that there may, and often do, exist many local optima for a given data set. This holds even if the data are sampled from a faithful probability distribution. Therefore, it is not wise to just run the learning algorithm once. Instead, we propose to run the algorithm a number of times on the same data and return the best CG found. However, there remains a question to answer for this approach to be applicable in practice: How do we know which of the learnt CGs is the best ? One option is selecting the CG learnt that has the best recall and/or precision w.r.t. the independencies detected by the hypothesis test in the sample set, rather than w.r.t. the separations in the CG sampled as we did in our experiments. Another option is scoring each learnt CG with a score such as the Bayesian information criterion (BIC). We expect that these scores correlate well with the recall and precision values in our experiments. Using a score inside the CKES algorithm instead of the hypothesis test is considered too time consuming due to the fact that there is no closed-form estimates of the maximum-likelihood parameters of a CG. However, we consider it feasible to use a score to rank the CGs learnt by multiple runs of CKES. We are currently working on these two alternative ways to make our approach to CG structure learning applicable in practice.

## Acknowledgments

This work is funded by the Center for Industrial Information Technology (CENIIT) and a so-called career contract at Linköping University, by the Swedish Research Council (ref. 2010-4808), and by FEDER funds and the Spanish Government (MICINN) through the project TIN2010-20900-C04-03.

## References

[1] D. M. Chickering. Optimal Structure Identification with Greedy Search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[2] M. Drton. Discrete Chain Graph Models. *Bernoulli*, 15:736–753, 2009.

[3] M. Drton and M. D. Perlman. A SINful Approach to Gaussian Graphical Model Selection. *Journal of Statistical Planning and Inference*, 138:1179–1200, 2008.

[4] S. L. Lauritzen. *Graphical Models.* Clarendon Press, Oxford, 1996.

[5] S. L. Lauritzen and N. Wermuth. Graphical Models for Association Between Variables, Some of Which are Qualitative and Some Quantitative. *The Annals of Statistics*, 17:31–57, 1989.

[6] Z. Ma, X. Xie, and Z. Geng. Structural Learning of Chain Graphs via Decomposition. *Journal of Machine Learning Research*, 9:2847–2880, 2008.

[7] C. Meek. *Graphical Models: Selecting Causal and Statistical Models.* PhD thesis, Carnegie Mellon University, 1997.

[8] J. D. Nielsen, T. Kočka, and J. M. Peña. On Local Optima in Learning Bayesian Networks. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 435–442, 2003.

[9] J. M. Peña. Faithfulness in Chain Graphs: The Discrete Case. *International Journal of Approximate Reasoning*, 50:1306–1313, 2009.

[10] J. M. Peña. Faithfulness in Chain Graphs: The Gaussian Case. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 588–599, 2011.

[11] M. Studený. On Recovery Algorithms for Chain Graphs. *International Journal of Approximate Reasoning*, 17:265–293, 1997.

[12] M. Studený. *Probabilistic Conditional Independence Structures.* Springer, 2005.

[13] M. Studený and R. R. Bouckaert. On Chain Graph Models for Description of Conditional Independence Structures. *The Annals of Statistics*, 26:1434–1495, 1998.

[14] M. Studený, A. Roverato, and Š. Štěpánová. Two Operations of Merging and Splitting Components in a Chain Graph. *Kybernetika*, 45:208–248, 1997.

[15] M. Volf and M. Studený. A Graphical Characterization of the Largest Chain Graphs. *International Journal of Approximate Reasoning*, 20:209–236, 1999.

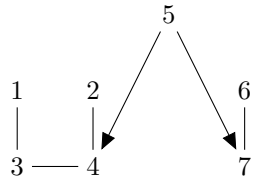
# Appendix of "An Inclusion Optimal Algorithm for Chain Graph Structure Learning"

J. M. Peña [a], Dag Sonntag [a] and Jens D. Nielsen [b]

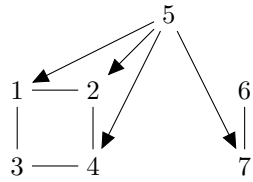
[a] ADIT, IDA, Linköping University, Sweden
[b] CLC bio, a Quiagen company, Denmark

## Appendix A: Example run

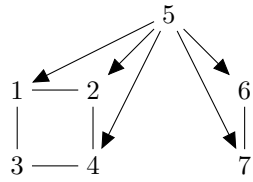
We show below an example run of the operation Fbsplit($K$, $L$, $G$) presented in Figure 1 of the main text. Let $G$ be the CG below, $K = \{1, 2, 3, 4, 6, 7\}$, and $L = \{3, 4, 7\}$.



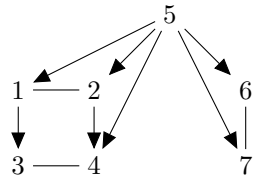

According to line 1, $L_1 = \{3, 4\}$ and $L_2 = \{7\}$. After having executed lines 2-4 for $i = 1$, $G$ looks like the CG below.



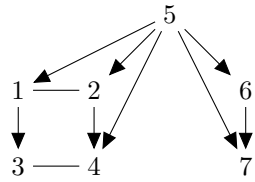

After having executed lines 2-4 for $i = 2$, $G$ looks like the CG below.



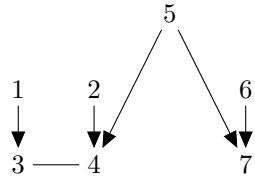

After having executed lines 5-8 for $i = 1$, $G$ looks like the CG below. Note that according to line 6, $K_j = \{1, 2, 3, 4\}$.



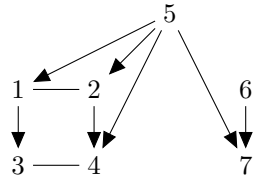

After having executed lines 5-8 for $i = 2$, $G$ looks like the CG below. Note that according to line 6, $K_j = \{6, 7\}$.



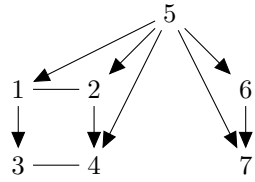

We show below an example run of the operation Fbmerge($L$, $R$, $G$) presented in Figure 1 of the main text. Let $G$ be the CG below, and $L = \{1, 2, 6\}$, and $R = \{3, 4, 7\}$.



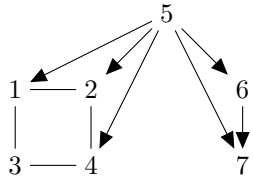

According to line 1, $R_1 = \{3, 4\}$ and $R_2 = \{7\}$. After having executed lines 2-4 for $i = 1$, $G$ looks like the CG below.



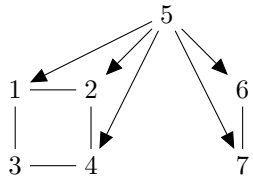

After having executed lines 2-4 for $i = 2$, $G$ looks like the CG below.



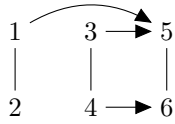
After having executed lines 5-8 for $i = 1$, $G$ looks like the CG below. Note that according to line 6, $L_j = \{1, 2\}$.




After having executed lines 5-8 for $i = 2$, $G$ looks like the CG below. Note that according to line 6, $L_j = \{6\}$.

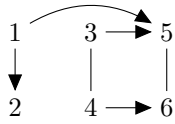



We show below an example run of the operation Construct $\beta(G, \alpha, \beta)$ presented in Figure 2 of the main text. Let $G$ be the CG below and $\alpha = (\{1\}, \{5, 6\}, \{3, 4\}, \{2\})$.

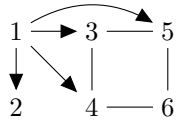



Initially, $H = G$. In the first iteration of the algorithm, $C = \{5, 6\}$, $\beta = (\{5, 6\})$, and the nodes $\{5, 6\}$ get removed from $H$. The algorithm jumps to line 3. In the second iteration, $C = \{3, 4\}$, $\beta = (\{3, 4\}, \{5, 6\})$, and the nodes $\{3, 4\}$ get removed from $H$. The algorithm jumps to line 3. In the third iteration, $C = \{1, 2\}$, $\beta = (\{1, 2\}, \{3, 4\}, \{5, 6\})$, and the nodes $\{1, 2\}$ get removed from $H$. The algorithm halts because $H = \varnothing$.

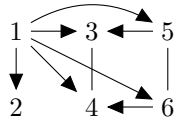
We finally show below an example run of the algorithm Method B3($G$, $\alpha$, $\beta$) presented in Figure 2 of the main text. Let $G$ be the CG above and $\alpha = (\{1\}, \{5, 6\}, \{3, 4\}, \{2\})$. As shown above, $\beta = (\{1, 2\}, \{3, 4\}, \{5, 6\})$ after having executed line 1. In the first iteration of the algorithm, $C = \{2\}$, $K = \{1, 2\}$, $G$ gets modified into the CG below by line 6 with $L = \{2\}$



and $\beta = (\{1\},\{2\},\{3,4\},\{5,6\})$ by line 7, $G$ does not get modified by line 10 but $\beta = (\{1\},\{2,3,4\},\{5,6\})$ by line 11. The algorithm jumps to line 3. In the second iteration, $C = \{2\}$, $K = \{2,3,4\}$, $G$ does not get modified by line 6 but $\beta = (\{1\},\{3,4\},\{2\},\{5,6\})$ by line 7, $G$ does not get modified by line 10 but $\beta = (\{1\},\{3,4\},\{2,5,6\})$ by line 11. The algorithm jumps to line 3. In the third iteration, $C = \{2\}$, $K = \{2,5,6\}$, $G$ does not get modified by line 6 but $\beta = (\{1\},\{3,4\},\{5,6\},\{2\})$ by line 7. The algorithm jumps to line 2. In the fourth iteration, $C = \{3,4\}$, $K = \{3,4\}$, $G$ gets modified into the CG below by line 10 with $L = \{3,4\}$ and $R = \{5,6\}$




and $\beta = (\{1\},\{3,4,5,6\},\{2\})$ by line 11. The algorithm jumps to line 3. In the fifth iteration, $C = \{3,4\}$, $K = \{3,4,5,6\}$, $G$ gets modified into the CG below by line 6 with $L = \{3,4\}$




and $\beta = (\{1\},\{5,6\},\{3,4\},\{2\})$ by line 7. The algorithm halts because $\beta = \alpha$.

# Appendix B: Proofs

**Lemma 1.** *Let $M$ denote an independence model, and $\alpha$ a chain $C_1,\ldots,C_n$. If $M$ is a graphoid, then there exits a unique CG $G_\alpha$ that is a MI map of $M$ relative to $\alpha$. Specifically, for each node $X$ of each block $C_k$ of $\alpha$, $Bd_{G_\alpha}(X)$ is the smallest subset $B$ of $\cup_{j=1}^{k} C_j \setminus \{X\}$ s.t. $X \perp_M \cup_{j=1}^{k} C_j \setminus \{X\} \setminus B | B$.*[1]

*Proof.* Let $X$ and $Y$ denote any two non-adjacent nodes of $G_\alpha$. Let $k^*$ denote the smallest $k$ s.t. $X, Y \in \cup_{j=1}^{k} C_j$. Assume without loss of generality that $X \in C_{k^*}$. Then, $X \perp_M \cup_{j=1}^{k^*} C_j \setminus \{X\} \setminus Bd_{G_\alpha}(X) | Bd_{G_\alpha}(X)$ by construction of $G_\alpha$ and, thus, $X \perp_M Y | \cup_{j=1}^{k^*} C_j \setminus \{X,Y\}$ by weak union. Then, $G_\alpha$ satisfies the pairwise block-recursive Markov property w.r.t. $M$ and, thus, $G_\alpha$ is an I map of $M$. In fact, $G_\alpha$ is a MI map of $M$ by construction of $Bd_{G_\alpha}(X)$.

Assume to the contrary that there exists another CG $H_\alpha$ that is a MI map of $M$ relative to $\alpha$. Let $X$ denote any node s.t. $Bd_{G_\alpha}(X) \neq Bd_{H_\alpha}(X)$. Let $X \in C_k$. Then, $X \perp_M \cup_{j=1}^{k} C_j \setminus \{X\} \setminus Bd_{G_\alpha}(X) | Bd_{G_\alpha}(X)$

[1] By convention, $X \perp_M \emptyset | \cup_{j=1}^{k} C_j \setminus \{X\}$.

and $X \perp_M \cup_{j=1}^{k} C_j \setminus \{X\} \setminus Bd_{H_\alpha}(X) | Bd_{H_\alpha}(X)$ because $G_\alpha$ and $H_\alpha$ are MI maps of $M$. Then, $X \perp_M \cup_{j=1}^{k} C_j \setminus \{X\} \setminus Bd_{G_\alpha}(X) \cap Bd_{H_\alpha}(X) | Bd_{G_\alpha}(X) \cap Bd_{H_\alpha}(X)$ by intersection. However, this contradicts the construction of $Bd_{G_\alpha}(X)$, because $Bd_{G_\alpha}(X) \cap Bd_{H_\alpha}(X)$ is smaller than $Bd_{G_\alpha}(X)$.

□

**Lemma 2.** *Let $G$ and $H$ denote two CGs s.t. $I(H) \subseteq I(G)$. For any component $C$ of $G$, there exists a unique component of $H$ that is maximal in $H$ from the set of components of $H$ that contain a descendant of $C$ in $G$.*

*Proof.* By definition of CG, there exists at least one such component of $H$. Assume to the contrary that there exist two such components of $H$, say $K$ and $K'$. Note that $Pa_H(K) \cap K' = \emptyset$ and $Pa_H(K') \cap K = \emptyset$ by definition of $K$ and $K'$. Note also that no node of $K$ or $Pa_H(K)$ is a descendant of $K'$ in $H$ by definition of $K$. This implies that $K' \perp_H K \cup Pa_H(K) \setminus Pa_H(K') | Pa_H(K')$ and, thus, $K \perp_H K' | Pa_H(K) \cup Pa_H(K')$ by weak union and symmetry.

That $K$ and $K'$ contain some descendants $k$ and $k'$ of $C$ in $G$ implies that there are descending routes from $C$ to $k$ and $k'$ in $G$ s.t. the nodes in the routes are descendant of $C$ in $G$. Thus, there is a route between $k$ and $k'$ in $G$ s.t. the nodes in the route are descendant of $C$ in $G$. Note that no node in this route is in $Pa_H(K)$ or $Pa_H(K')$ by definition of $K$ and $K'$. Then, $K \not\perp_G K' | Pa_H(K) \cup Pa_H(K')$. However, this contradicts the fact that $I(H) \subseteq I(G)$ because, as shown, $K \perp_H K' | Pa_H(K) \cup Pa_H(K')$.

□

**Lemma 3.** *Let $G$ and $H$ denote two CGs s.t. $I(H) \subseteq I(G)$. Let $\alpha$ denote a chain that is consistent with $H$. If no descendant of a node $X$ in $G$ is to the left of $X$ in $\alpha$, then the descendants of $X$ in $G$ are descendant of $X$ in $H$ too.*

*Proof.* Let $D$ denote the descendants of $X$ in $G$. Let $C$ denote the component of $G$ that contains $X$. Note that the descendants of $C$ in $G$ are exactly the set $D$. Then, there exists a unique component of $H$ that is maximal in $H$ from the set of components of $H$ that contain a node from $D$, by Lemma 2.

Let $K$ denote the component of $H$ that contains $X$. Note that $K$ is a component of $H$ that is maximal in $H$ from the set of components of $H$ that contain a node from $D$, since no node of $D$ is to the left of $X$ in $\alpha$. It follows from the paragraph above that $K$ is the only such component of $H$.

□

**Lemma 4.** *Let $G_\alpha$ denote the MI map of the independence model induced by a CG $G$ relative to a chain $\alpha$. Then, Method B3($G$, $\alpha$) returns $G_\alpha$.*

*Proof.* We start by proving that Method B3 halts at some point. When Method B3 is done with the rightmost block of $\alpha$, the rightmost block of $\beta$

contains all and only the nodes of the rightmost block of $\alpha$. When Method B3 is done with the second rightmost block of $\alpha$, the rightmost block of $\beta$ contains all and only the nodes of the rightmost block of $\alpha$, whereas the second rightmost block of $\beta$ contains all and only the nodes of the second rightmost block of $\alpha$. Continuing with this reasoning, one can see that when Method B3 is done with all the blocks of $\alpha$, $\beta$ coincides with $\alpha$ and thus Method B3 halts.

That Method B3 halts at some point implies that it performs a finite sequence of $m$ modifications to $G$ due to the fbsplit and fbmerging in lines 6 and 10. Let $G_t$ denote the CG resulting from the first $t$ modifications to $G$, and let $G_0 = G$. Specifically, Method B3 constructs $G_{t+1}$ from $G_t$ by either

- adding an edge $X - Y$ due to line 3 of Fbsplit or Fbmerge,
- adding an edge $X \rightarrow Y$ due to line 4 of Fbsplit or Fbmerge,
- performing all the component splits due to lines 5-8 of Fbsplit, or
- performing all the component mergings due to lines 5-8 of Fbmerge.

Note that none of the modifications above introduces new separation statements. This is trivial to see for the first and second modification. To see it for the third and fourth modification, recall that the splits and the mergings are part of a fbsplit and a fbmerging respectively and, thus, they are feasible. Therefore, $I(G_{t+1}) \subseteq I(G_t)$ for all $0 \leq t < m$ and, thus, $I(G_m) \subseteq I(G_0)$.

We continue by proving that $G_t$ is consistent with $\beta$ for all $0 \leq t \leq m$. Since this is true for $G_0$ due to line 1, it suffices to prove that if it is true for $G_t$ then it is true for $G_{t+1}$ for all $0 \leq t < m$. We consider the following four cases.

**Case 1** Method B3 constructs $G_{t+1}$ from $G_t$ by adding an edge $X - Y$ due to line 3 of Fbsplit or Fbmerge. It suffices to note that $X$ and $Y$ are in the same block of $G_t$ and $\beta$.

**Case 2** Method B3 constructs $G_{t+1}$ from $G_t$ by adding an edge $X \rightarrow Y$ due to line 4 of Fbsplit. It suffices to note that $X$ is to the left of $Y$ in $\beta$, because $G_t$ is consistent with $\beta$.

**Case 3** Method B3 constructs $G_{t+1}$ from $G_t$ by adding an edge $X \rightarrow Y$ due to line 4 of Fbmerge. Note that $X$ is to the left of $R$ in $\beta$, because $\beta$ is consistent with $G_t$. Then, $X$ is to the left of $L$ in $\beta$, because $L$ is the left neighbor of $R$ in $\beta$ and $X \notin L$. Then, $X$ is to the left of $Y$ in $\beta$, because $Y \in L$.

**Case 4** Method B3 constructs $G_{t+1}$ from $G_t$ by either performing all the component splits due to lines 5-8 of Fbsplit or performing all the component mergings due to lines 5-8 of Fbmerge. Note that the splits

and the mergings are feasible, since they are part of a fbsplit and a fbmerging respectively. Therefore, $G_{t+1}$ is a CG. Moreover, note that $\beta$ is modified immediately after the fbsplit and the fbmerging so that it is consistent with $G_{t+1}$.

Note that $G_m$ is not only consistent with $\beta$ but also with $\alpha$ because, as shown, $\beta$ coincides with $\alpha$ when Method B3 halts. In order to prove the lemma, i.e. that $G_m = G_\alpha$, all that remains to prove is that $I(G_\alpha) \subseteq I(G_m)$. To see it, note that $G_m = G_\alpha$ follows from $I(G_\alpha) \subseteq I(G_m)$, $I(G_m) \subseteq I(G_0)$, the fact that $G_m$ is consistent with $\alpha$, and the fact that $G_\alpha$ is the unique MI map of $I(G_0)$ relative to $\alpha$. Recall that $G_\alpha$ is guaranteed to be unique by Lemma 1, because $I(G_0)$ is a graphoid.

The rest of the proof is devoted to prove that $I(G_\alpha) \subseteq I(G_m)$. Specifically, we prove that if $I(G_\alpha) \subseteq I(G_t)$ then $I(G_\alpha) \subseteq I(G_{t+1})$ for all $0 \leq t < m$. Note that this implies that $I(G_\alpha) \subseteq I(G_m)$ because $I(G_\alpha) \subseteq I(G_0)$ by definition of MI map. First, we prove it when Method B3 constructs $G_{t+1}$ from $G_t$ by either performing all the component splits due to lines 5-8 of Fbsplit or performing all the component mergings due to lines 5-8 of Fbmerge. Note that the splits and the mergings are feasible, since they are part of a fbsplit and a fbmerging respectively. Therefore, $I(G_{t+1}) = I(G_t)$. Thus, $I(G_\alpha) \subseteq I(G_{t+1})$ because $I(G_\alpha) \subseteq I(G_t)$.

Now, we prove that if $I(G_\alpha) \subseteq I(G_t)$ then $I(G_\alpha) \subseteq I(G_{t+1})$ when Method B3 constructs $G_{t+1}$ from $G_t$ by adding a directed or undirected edge due to lines 3 and 4 of Fbsplit and Fbmerge. Specifically, we prove that if there is an $S$-active route $\rho_{t+1}^{AB}$ between two nodes $A$ and $B$ in $G_{t+1}$, then there is an $S$-active route between $A$ and $B$ in $G_\alpha$. We prove this result by induction on the number of occurrences of the added edge in $\rho_{t+1}^{AB}$. We assume without loss of generality that the added edge occurs in $\rho_{t+1}^{AB}$ as few or fewer times than in any other $S$-active route between $A$ and $B$ in $G_{t+1}$. We call this the minimality property of $\rho_{t+1}^{AB}$. If the number of occurrences of the added edge in $\rho_{t+1}^{AB}$ is zero, then $\rho_{t+1}^{AB}$ is an $S$-active route between $A$ and $B$ in $G_t$ too and, thus, there is an $S$-active route between $A$ and $B$ in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. Assume as induction hypothesis that the result holds for up to $n$ occurrences of the added edge in $\rho_{t+1}^{AB}$. We now prove it for $n+1$ occurrences. We consider the following four cases.

**Case 1** Method B3 constructs $G_{t+1}$ from $G_t$ by adding an edge $X - Y$ due to line 3 of Fbsplit. Note that $X - Y$ occurs in $\rho_{t+1}^{AB}$.[2] Assume that $X - Y$ occurs in a collider section of $\rho_{t+1}^{AB}$. Note that $X$ and $Y$ must be in the same component of $G_t$ for line 3 of Fbsplit to add an edge $X - Y$. This component also contains a node $Z$ that is in $S$ because, otherwise, $\rho_{t+1}^{AB}$ would not be $S$-active in $G_{t+1}$.[3] Note that there is a route $X - \ldots - Z - \ldots - Y$ in $G_t$. Then, we can replace any occurrence of $X - Y$ in a collider section of $\rho_{t+1}^{AB}$ with $X - \ldots - Z - \ldots - Y$, and thus

[2] Note that maybe $A = X$ and/or $Y = B$.

[3] Note that maybe $Z = X$ or $Z = Y$.

construct an $S$-active route between $A$ and $B$ in $G_{t+1}$ that violates the minimality property of $\rho^{AB}_{t+1}$. Since this is a contradiction, $X - Y$ only occurs in non-collider sections of $\rho^{AB}_{t+1}$. Let $\rho^{AB}_{t+1} = \rho^{AX}_{t+1} \cup X - Y \cup \rho^{YB}_{t+1}$. Note that $X, Y \notin S$ because, otherwise, $\rho^{AB}_{t+1}$ would not be $S$-active in $G_{t+1}$. For the same reason, $\rho^{AX}_{t+1}$ and $\rho^{YB}_{t+1}$ are $S$-active in $G_{t+1}$. Then, there are $S$-active routes $\rho^{AX}_{\alpha}$ and $\rho^{YB}_{\alpha}$ between $A$ and $X$ and between $Y$ and $B$ in $G_\alpha$ by the induction hypothesis.

Let $X - X' - \ldots - Y' - Y$ be a route in $G_t$ s.t. the nodes in $X' - \ldots - Y'$ are in $L$.[4] Such a route must exist for line 3 of Fbsplit to add an edge $X - Y$. Note that $X$ and $X'$ are adjacent in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. In fact, $X \to X'$ is in $G_\alpha$. To see it, recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Then, $K$ only contains nodes from $C$ or from blocks to the left of $C$ in $\alpha$. However, $X \notin C$ because $X \in K \setminus L$ and $L = K \cap C$. Then, $X$ is to the left of $C$ in $\alpha$. Thus, $X \to X'$ is in $G_\alpha$ because $X' \in L \subseteq C$. Likewise, $Y \to Y'$ is in $G_\alpha$. Note also that $X' - \ldots - Y'$ is in $G_\alpha$. To see it, note that the adjacencies in $X' - \ldots - Y'$ are preserved in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. Moreover, these adjacencies correspond to undirected edges in $G_\alpha$, because the nodes in $X' - \ldots - Y'$ are in $L$ and thus in the same block of $G_\alpha$, since $L \subseteq C$. Furthermore, a node in $X' - \ldots - Y'$ is in $S$ because, otherwise, $\rho^{AX}_{t+1} \cup X - X' - \ldots - Y' - Y \cup \rho^{YB}_{t+1}$ would be an $S$-active route between $A$ and $B$ in $G_{t+1}$ that would violate the minimality property of $\rho^{AB}_{t+1}$. Then, $\rho^{AX}_{\alpha} \cup X \to X' - \ldots - Y' \leftarrow Y \cup \rho^{YB}_{\alpha}$ is an $S$-active route between $A$ and $B$ in $G_\alpha$.

**Case 2** Method B3 constructs $G_{t+1}$ from $G_t$ by adding an edge $X \to Y$ due to line 4 of Fbsplit. Note that $X \to Y$ occurs in $\rho^{AB}_{t+1}$.[5] Assume that $X \to Y$ occurs as a collider edge in $\rho^{AB}_{t+1}$, i.e. $X \to Y$ occurs in a subroute of $\rho^{AB}_{t+1}$ of the form $X \to Y - \ldots - Z \leftarrow W$.[6] Note that a node in $Y - \ldots - Z$ is in $S$ because, otherwise, $\rho^{AB}_{t+1}$ would not be $S$-active in $G_{t+1}$. Let $X \to X' - \ldots - Y' - Y$ be a route in $G_t$ s.t. the nodes in $X' - \ldots - Y'$ are in $L$.[7] Such a route must exist for line 4 of Fbsplit to add an edge $X \to Y$. Then, we can replace $X \to Y - \ldots - Z \leftarrow W$ with $X \to X' - \ldots - Y' - Y - \ldots - Z \leftarrow W$ in $\rho^{AB}_{t+1}$, and thus construct an $S$-active route between $A$ and $B$ in $G_{t+1}$ that violates the minimality property of $\rho^{AB}_{t+1}$. Since this is a contradiction, $X \to Y$ never occurs as a collider edge in $\rho^{AB}_{t+1}$. Let $\rho^{AB}_{t+1} = \rho^{AX}_{t+1} \cup X \to Y \cup \rho^{YB}_{t+1}$. Note that $X, Y \notin S$ because, otherwise, $\rho^{AB}_{t+1}$ would not be $S$-active in $G_{t+1}$. For the same reason, $\rho^{AX}_{t+1}$ and $\rho^{YB}_{t+1}$ are $S$-active in $G_{t+1}$. Then, there are $S$-active routes $\rho^{AX}_{\alpha}$ and $\rho^{YB}_{\alpha}$ between $A$ and $X$ and between $Y$ and $B$ in $G_\alpha$ by the induction hypothesis.

[4] Note that maybe $X' = Y'$.
[5] Note that maybe $A = X$ and/or $Y = B$.
[6] Note that maybe $Y = Z$ and/or $W = X$.
[7] Note that maybe $X' = Y'$.

Let $X \rightarrow X' - \ldots - Y' - Y$ denote a route in $G_t$ s.t. the nodes in $X' - \ldots - Y'$ are in $L$.[8] Such a route must exist for line 4 of Fbsplit to add an edge $X \rightarrow Y$. Note that $X' - \ldots - Y'$ is in $G_\alpha$. To see it, note that the adjacencies in $X' - \ldots - Y'$ are preserved in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. Moreover, these adjacencies correspond to undirected edges in $G_\alpha$, because the nodes in $X' - \ldots - Y'$ are in $L$ and thus in the same block of $G_\alpha$, since $L \subseteq C$. Furthermore, a node in $X' - \ldots - Y'$ is in $S$ because, otherwise, $\rho_{t+1}^{AX} \cup X \rightarrow X' - \ldots - Y' - Y \cup \rho_{t+1}^{YB}$ would be an $S$-active route between $A$ and $B$ in $G_{t+1}$ that would violate the minimality property of $\rho_{t+1}^{AB}$. Moreover, note that $X$ and $X'$ are adjacent in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. In fact, $X \rightarrow X'$ is in $G_\alpha$. To see it, recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Then, no block to the left of $K$ in $\beta$ has a node from $C$ or from a block to the right of $C$ in $\alpha$. Note that $X$ is to the left of $K$ in $\beta$, because $\beta$ is consistent with $G_t$. Thus, $X \rightarrow X'$ is in $G_\alpha$ since $X' \in L \subseteq C$. Likewise, note that $Y'$ and $Y$ are adjacent in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. In fact, $Y' \leftarrow Y$ is in $G_\alpha$. To see it, note that $K$ only contains nodes from $C$ or from blocks to the left of $C$ in $\alpha$. However, $Y \notin C$ because $Y \in K \setminus L$ and $L = K \cap C$. Then, $Y$ is to the left of $C$ in $\alpha$. Thus, $Y' \leftarrow Y$ is in $G_\alpha$ because $Y' \in L \subseteq C$. Then, $\rho_\alpha^{AX} \cup X \rightarrow X' - \ldots - Y' \leftarrow Y \cup \rho_\alpha^{YB}$ is an $S$-active route between $A$ and $B$ in $G_\alpha$.

**Case 3** Method B3 constructs $G_{t+1}$ from $G_t$ by adding an edge $X - Y$ due to line 3 of Fbmerge. Note that $X - Y$ occurs in $\rho_{t+1}^{AB}$. We consider two cases.

**Case 3.1** Assume that $X - Y$ occurs in a collider section of $\rho_{t+1}^{AB}$. Let $\rho_{t+1}^{AB} = \rho_{t+1}^{AZ} \cup Z \rightarrow X' - \ldots - X - Y - \ldots - Y' \leftarrow W \cup \rho_{t+1}^{WB}$.[9] Note that $Z, W \notin S$ because, otherwise, $\rho_{t+1}^{AB}$ would not be $S$-active in $G_{t+1}$. For the same reason, $\rho_{t+1}^{AZ}$ and $\rho_{t+1}^{WB}$ are $S$-active in $G_{t+1}$. Then, there are $S$-active routes $\rho_\alpha^{AZ}$ and $\rho_\alpha^{WB}$ between $A$ and $Z$ and between $W$ and $B$ in $G_\alpha$ by the induction hypothesis.

Let $R_i$ denote the component of $G_t$ in $R$ that Fbmerge is processing when the edge $X - Y$ gets added. Recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Then, $R_i$ only contains nodes from $C$ or from blocks to the left of $C$ in $\alpha$. In other words, $R_i \subseteq \cup_{j=1}^{k^*} C_j \setminus \{X, Y\}$ where $C_{k^*} = C$ (recall that $X, Y \in L \subseteq C$). Therefore, $X \not\perp_{G_t} Y | \cup_{j=1}^{k^*} C_j \setminus \{X, Y\}$ because $X$ and $Y$ must be in $Pa_{G_t}(R_i)$ for line 3 of Fbmerge to add an edge $X - Y$. Then, $X$ and $Y$ are adjacent in $G_\alpha$ because,

[8] Note that maybe $X' = Y'$.

[9] Note that maybe $A = Z$, $X' = X$, $Y' = Y$, $W = Z$ and/or $W = B$.

otherwise, $X \perp_{G_\alpha} Y | \cup_{j=1}^{k^*} C_j \setminus \{X, Y\}$ which would contradict that $I(G_\alpha) \subseteq I(G_t)$. In fact, $X - Y$ is in $G_\alpha$ because $X$ and $Y$ are in the same block of $\alpha$, since $X, Y \in L \subseteq C$.

Note that $X' - \ldots - X$ and $Y - \ldots - Y'$ are in $G_\alpha$. To see it, note that the adjacencies in $X' - \ldots - X$ and $Y - \ldots - Y'$ are preserved in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. Moreover, these adjacencies correspond to undirected edges in $G_\alpha$, because the nodes in $X' - \ldots - X$ and $Y - \ldots - Y'$ are in $L$ since $X, Y \in L$ and, thus, they are in the same block of $G_\alpha$ since $L \subseteq C$. Then, $X' - \ldots - X - Y - \ldots - Y'$ is in $G_\alpha$. Furthermore, a node in $X' - \ldots - X - Y - \ldots - Y'$ is in $S$ because, otherwise, $\rho_{t+1}^{AB}$ would not be $S$-active in $G_{t+1}$. Note also that $Z$ and $X'$ are adjacent in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. In fact, $Z \rightarrow X'$ is in $G_\alpha$. To see it, recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Then, no block to the left of $L$ in $\beta$ has a node from $C$ or from a block to the right of $C$ in $\alpha$. Note that $Z$ is to the left of $L$ in $\beta$, because $\beta$ is consistent with $G_t$. Thus, $Z \rightarrow X'$ is in $G_\alpha$ since $X' \in L \subseteq C$. Likewise, $Y' \leftarrow W$ is in $G_\alpha$. Then, $\rho_\alpha^{AZ} \cup Z \rightarrow X' - \ldots - X - Y - \ldots - Y' \leftarrow W \cup \rho_\alpha^{WB}$ is an $S$-active route between $A$ and $B$ in $G_\alpha$.

**Case 3.2** Assume that $X - Y$ occurs in a non-collider section of $\rho_{t+1}^{AB}$. Note that this implies that $G_t$ has a descending route from $X$ to $A$ or to a node in $S$, or from $Y$ to $B$ or to a node in $S$. Assume without loss of generality that $G_t$ has a descending route from $Y$ to $B$ or to a node in $S$.

Let $R_i$ denote the component of $G_t$ in $R$ that Fbmerge is processing when the edge $X - Y$ gets added. Let $L_Y$ denote the component of $G_t$ that contains the node $Y$. Let $D$ denote the component of $G_\alpha$ that is maximal in $G_\alpha$ from the set of components of $G_\alpha$ that contain a descendant of $L_Y$ in $G_t$. Recall that $D$ is guaranteed to be unique by Lemma 2, because $I(G_\alpha) \subseteq I(G_t)$. We now show that some $d \in D$ is a descendant of $R_i$ in $G_t$. We consider four cases.

**Case 3.2.1** Assume that $D \cap L_Y \neq \emptyset$. It suffices to consider any $d \in R_i$. To see it, recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Then, $R_i$ only contains nodes from $C$ or from blocks to the left of $C$ in $\alpha$. Thus, $d$ is not to the right of the nodes of $D \cap L_Y$ in $\alpha$, since $L_Y \subseteq L \subseteq C$. Moreover, $d$ is not to the left of the nodes of $D \cap L_Y$ in $\alpha$ because, otherwise, there would be a contradiction with the definition of $D$. Then, $d \in D$.

**Case 3.2.2** Assume that $D \cap L_Y = \emptyset$ and $D \cap R_i \neq \emptyset$. It suffices to consider any $d \in D \cap R_i$.

**Case 3.2.3** Assume that $D \cap L_Y = \emptyset$, $D \cap R_i = \emptyset$, and some $d \in D$ was a descendant of some $r \in R_i$ in $G_0$. Recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Then, $R_i$ only contains nodes from $C$ or from blocks to the left of $C$ in $\alpha$. Then, $r$ was not in the blocks of $\alpha$ previously considered, since $r \in R_i$. Therefore, no descendant of $r$ in $G_0$ is currently to the left of $r$ in $\beta$ and, thus, the descendants of $r$ in $G_0$ are descendant of $r$ in $G_t$ by Lemma 3, because $I(G_t) \subseteq I(G_0)$ and $\beta$ is consistent with $G_t$. Then, $d$ is a descendant of $r$ and thus of $R_i$ in $G_t$.

**Case 3.2.4** Assume that $D \cap L_Y = \emptyset$, $D \cap R_i = \emptyset$, and no node of $D$ was a descendant of a node of $R_i$ in $G_0$. As shown in Case 3.2.3, the descendants of any node $r \in R_i$ in $G_0$ are descendant of $r$ in $G_t$ too. Therefore, no descendant of $r$ in $G_0$ was to the left of the nodes of $D$ in $\alpha$ because, otherwise, a descendant of $r$ and thus of $L_Y$ in $G_t$ would be to the left of the nodes of $D$ in $\alpha$, which would contradict the definition of $D$. Recall that no descendant of $r$ in $G_0$ was in $D$ either. Note also that the nodes of $D$ are to the left of the nodes of $R_i$ in $\alpha$, by definition of $D$ and the fact that $D \cap R_i = \emptyset$. These observations have two consequences. First, the components of $G$ containing a node from $D$ were still in $H$ when any component of $G$ containing a node from $R_i$ became a terminal component of $H$ in Construct $\beta$. Thus, Construct $\beta$ added the components of $G$ containing a node from $D$ to $\beta$ after having added the components of $G$ containing a node from $R_i$. Second, Construct $\beta$ did not interchange in $\beta$ any component of $G$ containing a node from $D$ with any component of $G$ containing a node from $R_i$.

Recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Note that the nodes of $D$ were not in the blocks of $\alpha$ previously considered because, otherwise, $C$ and thus the nodes of $L_Y$ (recall that $L_Y \subseteq L \subseteq C$) would be to the left of $D$ in $\alpha$, which would contradict the definition of $D$. Therefore, the nodes of $D$ are currently still to the left of $R_i$ in $\beta$. Note that the only component to the left of $R_i$ in $\beta$ that contains a descendant of $L_Y$ in $G_t$ is precisely $L_Y$, because $L$ is the left neighbor of $R$ in $\beta$, $L_Y \subseteq L$, and $\beta$ is consistent with $G_t$. However, $D \cap L_Y = \emptyset$. Thus, $D$ contains no descendant of $L_Y$ in $G_t$, which contradicts the definition of $D$. Thus, this case never occurs.

We continue with the proof of Case 3.2. Let $\rho_{t+1}^{AB} = \rho_{t+1}^{AX} \cup X -$

$Y \cup \rho_{t+1}^{YB}$.[10] Note that $X, Y \notin S$ because, otherwise, $\rho_{t+1}^{AB}$ would not be $S$-active in $G_{t+1}$. For the same reason, $\rho_{t+1}^{AX}$ and $\rho_{t+1}^{YB}$ are $S$-active in $G_{t+1}$. Note that $X$ and $Y$ must be in $Pa_{G_t}(R_i)$ for line 3 of Fbmerge to add an edge $X - Y$. Then, no descendant of $R_i$ in $G_t$ is in $S$ because, otherwise, there would be an $S$-active route $\rho_t^{XY}$ between $X$ and $Y$ in $G_t$ and, thus, $\rho_{t+1}^{AX} \cup \rho_t^{XY} \cup \rho_{t+1}^{YB}$ would be an $S$-active route between $A$ and $B$ in $G_{t+1}$ that would violate the minimality property of $\rho_{t+1}^{AB}$. Then, there is an $S$-active descending route $\rho_t^{rd}$ from some $r \in R_i$ to some $d \in D$ in $G_t$ because, as shown, $D$ contains a descendant of $R_i$ in $G_t$. Then, $\rho_{t+1}^{AX} \cup X \to X' - \ldots - r \cup \rho_t^{rd}$ is an $S$-active route between $A$ and $d$ in $G_{t+1}$.[11] Likewise, $\rho_{t+1}^{BY} \cup Y \to Y' - \ldots - r \cup \rho_t^{rd}$ is an $S$-active route between $B$ and $d$ in $G_{t+1}$, where $\rho_{t+1}^{BY}$ denotes the route resulting from reversing $\rho_{t+1}^{YB}$.[12] Therefore, there are $S$-active routes $\rho_\alpha^{Ad}$ and $\rho_\alpha^{Bd}$ between $A$ and $d$ and between $B$ and $d$ in $G_\alpha$ by the induction hypothesis.

Recall that we assumed without loss of generality that $G_t$ has a descending route from $Y$ to a node $E$ s.t. $E = B$ or $E \in S$. Note that $E$ is a descendant of $L_Y$ in $G_t$ and, thus, $E$ is a descendant of $d$ in $G_\alpha$ by definition of $D$ and the fact that $d \in D$. Let $\rho_\alpha^{dE}$ denote the descending route from $d$ to $E$ in $G_\alpha$. Assume without loss of generality that $G_\alpha$ has no descending route from $d$ to $B$ or to a node of $S$ that is shorter than $\rho_\alpha^{dE}$. We now consider two cases.

**Case 3.2.5** Assume that $E = B$. Note that $\rho_\alpha^{dE}$ is $S$-active in $G_\alpha$ by definition and the fact that $d \notin S$. To see the latter, recall that no descendant of $R_i$ in $G_t$ (among which is $d$) is in $S$. Thus, $\rho_\alpha^{Ad} \cup \rho_\alpha^{dE}$ is an $S$-active route between $A$ and $B$ in $G_\alpha$.

**Case 3.2.6** Assume that $E \in S$. Let $\rho_\alpha^{dB}$ and $\rho_\alpha^{Ed}$ denote the routes resulting from reversing $\rho_\alpha^{Bd}$ and $\rho_\alpha^{dE}$. Consider the route $\rho_\alpha^{Ad} \cup \rho_\alpha^{dB}$ between $A$ and $B$ in $G_\alpha$. If this route is $S$-active, then we are done. If it is not $S$-active in $G_\alpha$, then $d$ occurs in a collider section of $\rho_\alpha^{Ad} \cup \rho_\alpha^{dB}$ that has no node in $S$. Then, we can replace each such occurrence of $d$ with $\rho_\alpha^{dE} \cup \rho_\alpha^{Ed}$ and, thus construct an $S$-active route between $A$ and $B$ in $G_\alpha$.

**Case 4** Method B3 constructs $G_{t+1}$ from $G_t$ by adding an edge $X \to Y$ due to line 4 of Fbmerge. Note that $X \to Y$ occurs in $\rho_{t+1}^{AB}$. We consider two cases.

[10] Note that maybe $A = X$ and/or $Y = B$.
[11] Note that maybe $X' = r$.
[12] Note that maybe $Y' = r$.

**Case 4.1** Assume that $X \to Y$ occurs as a collider edge in $\rho_{t+1}^{AB}$. Let $\rho_{t+1}^{AB} = \rho_{t+1}^{AX} \cup X \to Y \cup \rho_{t+1}^{YB}$.[13] Note that $X \notin S$ because, otherwise, $\rho_{t+1}^{AB}$ would not be $S$-active in $G_{t+1}$. For the same reason, $\rho_{t+1}^{AX}$ is $S$-active in $G_{t+1}$. Then, there is an $S$-active route $\rho_{\alpha}^{AX}$ between $A$ and $X$ in $G_\alpha$ by the induction hypothesis.

Let $R_i$ denote the component of $G_t$ in $R$ that Fbmerge is processing when the edge $X \to Y$ gets added. Recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Then, $R_i$ only contains nodes from $C$ or from blocks to the left of $C$ in $\alpha$. In other words, $R_i \subseteq \cup_{j=1}^{k^*} C_j \setminus \{X, Y\}$ where $k^*$ is the smallest $k$ s.t. $X, Y \in \cup_{j=1}^{k} C_j$ (recall that $Y \in L \subseteq C$). Therefore, $X \not\perp_{G_t} Y | \cup_{j=1}^{k^*} C_j \setminus \{X, Y\}$ because $X$ and $Y$ must be in $Pa_{G_t}(R_i)$ for line 4 of Fbmerge to add an edge $X \to Y$. Then, $X$ and $Y$ are adjacent in $G_\alpha$ because, otherwise, $X \perp_{G_\alpha} Y | \cup_{j=1}^{k^*} C_j \setminus \{X, Y\}$ which would contradict that $I(G_\alpha) \subseteq I(G_t)$. In fact, $X \to Y$ is in $G_\alpha$. To see it, recall that Method B3 is currently considering the block $C$ of $\alpha$, and that it has previously considered all the blocks of $\alpha$ to the right of $C$ in $\alpha$. Then, no block to the left of $L$ in $\beta$ has a node from $C$ or from a block to the right of $C$ in $\alpha$. Note that $X$ is to the left of $R$ in $\beta$, because $\beta$ is consistent with $G_t$. Then, $X$ is to the left of $L$ in $\beta$, because $L$ is the left neighbor of $R$ in $\beta$ and $X \notin L$. Thus, $X \to Y$ is in $G_\alpha$ because $Y \in L \subseteq C$. We now consider two cases.

**Case 4.1.1** Assume that $\rho_{t+1}^{YB} = Y - \ldots - Y \leftarrow X \cup \rho_{t+1}^{XB}$. Note that a node in $Y - \ldots - Y$ is in $S$ because, otherwise, $\rho_{t+1}^{AB}$ would not be $S$-active in $G_{t+1}$. For the same reason, $\rho_{t+1}^{XB}$ is $S$-active in $G_{t+1}$. Then, there is an $S$-active route $\rho_{\alpha}^{XB}$ between $X$ and $B$ in $G_\alpha$ by the induction hypothesis. Note that $Y - \ldots - Y$ is in $G_\alpha$. To see it, note that the adjacencies in $Y - \ldots - Y$ are preserved in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. Moreover, these adjacencies correspond to undirected edges in $G_\alpha$, because the nodes in $Y - \ldots - Y$ are in $L$ since $Y \in L$ and, thus, they are in the same block of $G_\alpha$ since $L \subseteq C$. Then, $\rho_{\alpha}^{AX} \cup X \to Y - \ldots - Y \leftarrow X \cup \rho_{\alpha}^{XB}$ is an $S$-active route between $A$ and $B$ in $G_\alpha$.

**Case 4.1.2** Assume that $\rho_{t+1}^{YB} = Y - \ldots - Z \leftarrow W \cup \rho_{t+1}^{WB}$.[14] Note that $W \notin S$ and a node in $Y - \ldots - Z$ is in $S$ because, otherwise, $\rho_{t+1}^{AB}$ would not be $S$-active in $G_{t+1}$. For the same reason, $\rho_{t+1}^{WB}$ is $S$-active in $G_{t+1}$. Then, there is an $S$-active route $\rho_{\alpha}^{WB}$ between $W$ and $B$ in $G_\alpha$ by the induction hypothesis.

[13] Note that maybe $A = X$ and/or $Y = B$.

[14] Note that maybe $Y = Z$, $W = X$ and/or $W = B$. Note that $Y \neq Z$ or $W \neq X$, because the case where $Y = Z$ and $W = X$ is covered by Case 4.1.1.

Note that $Y - \ldots - Z$ is in $G_\alpha$. To see it, note that the adjacencies in $Y - \ldots - Z$ are preserved in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. Moreover, these adjacencies correspond to undirected edges in $G_\alpha$, because the nodes in $Y - \ldots - Z$ are in $L$ since $Y \in L$ and, thus, they are in the same block of $G_\alpha$ since $L \subseteq C$. Moreover, note that $Z$ and $W$ are adjacent in $G_\alpha$ since $I(G_\alpha) \subseteq I(G_t)$. In fact, $Z \leftarrow W$ is in $G_\alpha$. To see it, recall that no block to the left of $L$ in $\beta$ has a node from $C$ or from a block to the right of $C$ in $\alpha$. Note that $W$ is to the left of $L$ in $\beta$, because $\beta$ is consistent with $G_t$. Thus, $Z \leftarrow W$ is in $G_\alpha$ since $Z \in L \subseteq C$. Then, $\rho_\alpha^{AX} \cup X \rightarrow Y - \ldots - Z \leftarrow W \cup \rho_\alpha^{WB}$ is an $S$-active route between $A$ and $B$ in $G_\alpha$.

**Case 4.2** Assume that $X \rightarrow Y$ occurs as a non-collider edge in $\rho_{t+1}^{AB}$. The proof of this case is the same as that of Case 3.2, with the only exception that $X - Y$ should be replaced by $X \rightarrow Y$.

□

**Theorem 1.** *Given two CGs $G$ and $H$ s.t. $I(H) \subseteq I(G)$, Method G2H(G,H) transforms $G$ into $H$ by a sequence of directed and undirected edge additions and feasible splits and mergings s.t. after each operation in the sequence $G$ is a CG and $I(H) \subseteq I(G)$.*

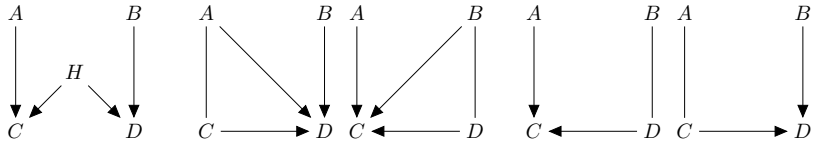
*Proof.* Note from line 1 that $\alpha$ denotes a chain that is consistent with $H$. Let $G_\alpha$ denote the MI map of $I(G)$ relative to $\alpha$. Recall that $G_\alpha$ is guaranteed to be unique by Lemma 1, because $I(G)$ is a graphoid. Note that $I(H) \subseteq I(G)$ implies that $G_\alpha$ is a subgraph of $H$. To see it, note that $I(H) \subseteq I(G)$ implies that we can obtain a MI map of $I(G)$ relative to $\alpha$ by just removing edges from $H$. However, $G_\alpha$ is the only MI map of $I(G)$ relative to $\alpha$.

Then, it follows from the proof of Lemma 4 that line 2 transforms $G$ into $G_\alpha$ by a sequence of directed and undirected edge additions and feasible splits and mergings, and that after each operation in the sequence $G$ is a CG and $I(G_\alpha) \subseteq I(G)$. Thus, after each operation in the sequence $I(H) \subseteq I(G)$ because $I(H) \subseteq I(G_\alpha)$ since, as shown, $G_\alpha$ is a subgraph of $H$. Finally, line 3 transforms $G$ from $G_\alpha$ to $H$ by a sequence of edge additions. Of course, after each edge addition $G$ is a CG and $I(H) \subseteq I(G)$ because $G_\alpha$ is a subgraph of $H$.

□

**Theorem 2.** *For any probability distribution $p$ for which the composition property holds, the CKES algorithm finds a CG that is inclusion optimal w.r.t. $p$.*

*Proof.* Let $R$, $S$ and $T$ be three random variables. Let $H(R|S)$ denote the conditional entropy of $R$ given $S$. Then, $H(R|S,T) \leq H(R|S)$ and, moreover, $H(R|S,T) = H(R|S)$ iff $R \perp_p T|S$ [2, Chapter 2]. Therefore,




The $w$-structure CG (a) Inclusion optimal CGs (b) CGs learnt by the LCD algorithm (c)

Figure 1: CGs In The Experiment.

Table 1: Average Results For Gaussian Data.

| CG structure | Recall | | Precision | |
|---|---|---|---|---|
| | CKES | LCD | CKES | LCD |
| n=100 | 0.54 | 0.61 | 0.75 | 0.44 |
| n=30000 | 0.57 | 0.57 | 1.00 | 0.57 |

$H(Y|Bd_G(Y))$ stays the same when removing the edge $X \to Y$ from $G$ in line 4. Likewise, $H(X|Bd_G(X))$ and $H(Y|Bd_G(Y))$ stay the same when removing $X - Y$ from $G$ in line 5. On the other hand, $H(Y|Bd_G(Y))$ decreases when adding the edge $X \to Y$ to $G$ in line 6. Likewise, $H(X|Bd_G(X))$ or $H(Y|Bd_G(Y))$ decreases when adding the edge $X - Y$ to $G$ in line 7. Let us define the score of a CG $G$ as $\sum_{X \in V} H(X|Bd_G(X))$. Now, note that the algorithm cannot enter an endless loop, i.e. it cannot perform a sequence of edge additions and removals so that the CGs before and after the sequence coincide. To see it, assume the contrary and note that such a sequence must contain both edge additions and removals. However, such a sequence would imply that the score of the CG after the sequence is smaller than the score of the CG before the sequence, which is a contradiction. If the algorithm cannot enter an endless loop, then the algorithm must reach a CG $G$ such that no edge can be added or removed from it in lines 3-7, because the number of CGs is finite. Moreover, $I(G) \subseteq I(p)$. To see it, assume the contrary. Then, according to the local Markov property and the composition property, there must exist two nodes $X$ and $Y$ such that $X \notin Sd_G(Y) \cup Bd_G(Y)$, $X \not\perp_p Y|Bd_G(Y)$ but $X \perp_G Y|Bd_G(Y)$. Then, a new edge can be added to $G$ in line 6 or 7, which is a contradiction. Obviously, $I(G) \subseteq I(p)$ still holds after $G$ has been updated in line 8. Therefore, the next execution of lines 3-7 may remove edges from $G$ but it never adds edges to $G$. Consequently, the algorithm must reach a CG $G$ such that no edge can be removed from it, because the number of edges that can be removed is finite. At this point, the algorithm executes line 8 repeatedly. This implies that the algorithm terminates at some point, because any CG can be transformed in any other equivalent CG via a sequence of feasible merges and splits [3, Corollary 7].

Finally, we show that there is no model $G'$ such that $I(G) \subset I(G') \subseteq I(p)$

when the algorithm has terminated in $G$. Assume the contrary. Theorem 1 says that there must exist a model $H$ such that $I(G) = I(F) \subset I(H) \subseteq \ldots \subseteq I(G') \subseteq I(p)$ and $H$ can be reached by a single edge removal from a CG $F$ which is in the equivalence class of $G$.[15] However this edge must be between two nodes $X$ and $Y$ in $F$ such that $X \in Bd_F(Y)$ and $X \perp_p Y | Bd_F(Y) \setminus X$ because $I(H) \subseteq I(p)$. This now contradicts the assumption since we know that the algorithm only can terminate if there exists no such edge for any CG in the equivalence class of $G$. Hence, we have a contradiction.

□

# Appendix C: Additional experiment

In this experiment, we want to show how the algorithms handle an unfaithful probability distribution that satisfies the composition property. [1] performed a similar experiment and we used the same setup. Specifically, we used the $w$-structure seen in Figure 1a as CG structure. From this structure, 10 Gaussian probability distributions were generated and sampled into sample sets in the same way as we did for the experiment in our paper. We then removed the $H$ variable from the sample sets. This meant that the algorithms tried to model the (in)dependencies in Figure 1a with only $A$, $B$, $C$ and $D$ as nodes. It can be shown that this is impossible (i.e. that there exists no faithful CG for this probability distribution) and that the inclusion optimal CGs are those seen in Figure 1b [1]. As performance measures, we computed the recall of the separations over $\{A, B, C, D\}$ in the 5-node CG sampled, and the precision of the separations in the 4-node CGs learnt. This meant that it was impossible to find a CG with perfect precision and recall, but that the best inclusion optimal CG had precision 1 and recall 0.57. The results obtained are shown in Figure 1. The table show average results over the 10 sample sets in the experiment. The results for categorical data are similar and, thus, they are not included here.

Running the example above by hand with perfect information, it can be seen that the CKES algorithm will end up in one of the CGs seen in Figure 1b. The LCD algorithm on the other hand ends up in one of the CGs shown in Figure 1c, which do not include the original independence model. The experimental results obtained with large sample sets (30000 samples) are shown in Figure 1 and coincide with the theoretical results just described. For the smaller sample sets (100 samples), the same trend observed in the experiment in our paper can be seen here (i.e. the CKES algorithm learning a CG with higher precision and the LCD algorithm learning a CG with higher recall). Therefore, we reach here the same conclusion as we did previously: Since the goal of structural learning is to find a model representing an I map of the probability distribution at hand, the CKES algorithm achieves better results than the LCD algorithm.

[15] Note that $H$ might be equal to $G'$ and $F$ might be equal to $G$.

# References

[1] D. M. Chickering and C. Meek. Finding Optimal Bayesian Networks. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 94–102, 2002.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[3] M. Studený, A. Roverato, and Š. Štěpánová. Two Operations of Merging and Splitting Components in a Chain Graph. *Kybernetika*, 45:208–248, 1997.

Department of Computer and Information Science
Linköpings universitet

## Licentiate Theses

## Linköpings Studies in Science and Technology
## Faculty of Arts and Sciences

No 17 **Vojin Plavsic:** Interleaved Processing of Non-Numerical Data Stored on a Cyclic Memory. (Available at: FOA, Box 1165, S-581 11 Linköping, Sweden. FOA Report B30062E)

No 28 **Arne Jönsson, Mikael Patel:** An Interactive Flowcharting Technique for Communicating and Realizing Algorithms, 1984.

No 29 **Johnny Eckerland:** Retargeting of an Incremental Code Generator, 1984.

No 48 **Henrik Nordin:** On the Use of Typical Cases for Knowledge-Based Consultation and Teaching, 1985.

No 52 **Zebo Peng:** Steps Towards the Formalization of Designing VLSI Systems, 1985.

No 60 **Johan Fagerström:** Simulation and Evaluation of Architecture based on Asynchronous Processes, 1985.

No 71 **Jalal Maleki:** ICONStraint, A Dependency Directed Constraint Maintenance System, 1987.

No 72 **Tony Larsson:** On the Specification and Verification of VLSI Systems, 1986.

No 73 **Ola Strömfors:** A Structure Editor for Documents and Programs, 1986.

No 74 **Christos Levcopoulos:** New Results about the Approximation Behavior of the Greedy Triangulation, 1986.

No 104 **Shamsul I. Chowdhury: S**tatistical Expert Systems - a Special Application Area for Knowledge-Based Computer Methodology, 1987.

No 108 **Rober Bilos:** Incremental Scanning and Token-Based Editing, 1987.

No 111 **Hans Block:** SPORT-SORT Sorting Algorithms and Sport Tournaments, 1987.

No 113 **Ralph Rönnquist:** Network and Lattice Based Approaches to the Representation of Knowledge, 1987.

No 118 **Mariam Kamkar, Nahid Shahmehri:** Affect-Chaining in Program Flow Analysis Applied to Queries of Programs, 1987.

No 126 **Dan Strömberg:** Transfer and Distribution of Application Programs, 1987.

No 127 **Kristian Sandahl:** Case Studies in Knowledge Acquisition, Migration and User Acceptance of Expert Systems, 1987.

No 139 **Christer Bäckström:** Reasoning about Interdependent Actions, 1988.

No 140 **Mats Wirén:** On Control Strategies and Incrementality in Unification-Based Chart Parsing, 1988.

No 146 **Johan Hultman:** A Software System for Defining and Controlling Actions in a Mechanical System, 1988.

No 150 **Tim Hansen:** Diagnosing Faults using Knowledge about Malfunctioning Behavior, 1988.

No 165 **Jonas Löwgren:** Supporting Design and Management of Expert System User Interfaces, 1989.

No 166 **Ola Petersson:** On Adaptive Sorting in Sequential and Parallel Models, 1989.

No 174 **Yngve Larsson:** Dynamic Configuration in a Distributed Environment, 1989.

No 177 **Peter Åberg:** Design of a Multiple View Presentation and Interaction Manager, 1989.

No 181 **Henrik Eriksson:** A Study in Domain-Oriented Tool Support for Knowledge Acquisition, 1989.

No 184 **Ivan Rankin:** The Deep Generation of Text in Expert Critiquing Systems, 1989.

No 187 **Simin Nadjm-Tehrani:** Contributions to the Declarative Approach to Debugging Prolog Programs, 1989.

No 189 **Magnus Merkel:** Temporal Information in Natural Language, 1989.

No 196 **Ulf Nilsson:** A Systematic Approach to Abstract Interpretation of Logic Programs, 1989.

No 197 **Staffan Bonnier:** Horn Clause Logic with External Procedures: Towards a Theoretical Framework, 1989.

No 203 **Christer Hansson:** A Prototype System for Logical Reasoning about Time and Action, 1990.

No 212 **Björn Fjellborg:** An Approach to Extraction of Pipeline Structures for VLSI High-Level Synthesis, 1990.

No 230 **Patrick Doherty:** A Three-Valued Approach to Non-Monotonic Reasoning, 1990.

No 237 **Tomas Sokolnicki:** Coaching Partial Plans: An Approach to Knowledge-Based Tutoring, 1990.

No 250 **Lars Strömberg:** Postmortem Debugging of Distributed Systems, 1990.

No 253 **Torbjörn Näslund:** SLDFA-Resolution - Computing Answers for Negative Queries, 1990.

No 260 **Peter D. Holmes:** Using Connectivity Graphs to Support Map-Related Reasoning, 1991.

No 283 **Olof Johansson:** Improving Implementation of Graphical User Interfaces for Object-Oriented Knowledge- Bases, 1991.

No 298 **Rolf G Larsson:** Aktivitetsbaserad kalkylering i ett nytt ekonomisystem, 1991.

No 318 **Lena Strömbäck:** Studies in Extended Unification-Based Formalism for Linguistic Description: An Algorithm for Feature Structures with Disjunction and a Proposal for Flexible Systems, 1992.

No 319 **Mikael Pettersson:** DML-A Language and System for the Generation of Efficient Compilers from Denotational Specification, 1992.

No 326 **Andreas Kågedal:** Logic Programming with External Procedures: an Implementation, 1992.

No 328 **Patrick Lambrix:** Aspects of Version Management of Composite Objects, 1992.

No 333 **Xinli Gu:** Testability Analysis and Improvement in High-Level Synthesis Systems, 1992.

No 335 **Torbjörn Näslund:** On the Role of Evaluations in Iterative Development of Managerial Support Systems, 1992.

No 348 **Ulf Cederling:** Industrial Software Development - a Case Study, 1992.

No 352 **Magnus Morin:** Predictable Cyclic Computations in Autonomous Systems: A Computational Model and Implementation, 1992.

No 371 **Mehran Noghabai:** Evaluation of Strategic Investments in Information Technology, 1993.

No 378 **Mats Larsson:** A Transformational Approach to Formal Digital System Design, 1993.

No 380 **Johan Ringström:** Compiler Generation for Parallel Languages from Denotational Specifications, 1993.
No 381 **Michael Jansson:** Propagation of Change in an Intelligent Information System, 1993.
No 383 **Jonni Harrius:** An Architecture and a Knowledge Representation Model for Expert Critiquing Systems, 1993.
No 386 **Per Österling:** Symbolic Modelling of the Dynamic Environments of Autonomous Agents, 1993.
No 398 **Johan Boye:** Dependency-based Groudness Analysis of Functional Logic Programs, 1993.
No 402 **Lars Degerstedt:** Tabulated Resolution for Well Founded Semantics, 1993.
No 406 **Anna Moberg:** Satellitkontor - en studie av kommunikationsmönster vid arbete på distans, 1993.
No 414 **Peter Carlsson:** Separation av företagsledning och finansiering - fallstudier av företagsledarutköp ur ett agent-teoretiskt perspektiv, 1994.
No 417 **Camilla Sjöström:** Revision och lagreglering - ett historiskt perspektiv, 1994.
No 436 **Cecilia Sjöberg:** Voices in Design: Argumentation in Participatory Development, 1994.
No 437 **Lars Viklund:** Contributions to a High-level Programming Environment for a Scientific Computing, 1994.
No 440 **Peter Loborg:** Error Recovery Support in Manufacturing Control Systems, 1994.
FHS 3/94 **Owen Eriksson:** Informationssystem med verksamhetskvalitet - utvärdering baserat på ett verksamhetsinriktat och samskapande perspektiv, 1994.
FHS 4/94 **Karin Pettersson:** Informationssystemstrukturering, ansvarsfördelning och användarinflytande - En komparativ studie med utgångspunkt i två informationssystemstrategier, 1994.
No 441 **Lars Poignant:** Informationsteknologi och företagsetablering - Effekter på produktivitet och region, 1994.
No 446 **Gustav Fahl:** Object Views of Relational Data in Multidatabase Systems, 1994.
No 450 **Henrik Nilsson:** A Declarative Approach to Debugging for Lazy Functional Languages, 1994.
No 451 **Jonas Lind:** Creditor - Firm Relations: an Interdisciplinary Analysis, 1994.
No 452 **Martin Sköld:** Active Rules based on Object Relational Queries - Efficient Change Monitoring Techniques, 1994.
No 455 **Pär Carlshamre:** A Collaborative Approach to Usability Engineering: Technical Communicators and System Developers in Usability-Oriented Systems Development, 1994.
FHS 5/94 **Stefan Cronholm:** Varför CASE-verktyg i systemutveckling? - En motiv- och konsekvensstudie avseende arbetssätt och arbetsformer, 1994.
No 462 **Mikael Lindvall:** A Study of Traceability in Object-Oriented Systems Development, 1994.
No 463 **Fredrik Nilsson:** Strategi och ekonomisk styrning - En studie av Sandviks förvärv av Bahco Verktyg, 1994.
No 464 **Hans Olsén:** Collage Induction: Proving Properties of Logic Programs by Program Synthesis, 1994.
No 469 **Lars Karlsson:** Specification and Synthesis of Plans Using the Features and Fluents Framework, 1995.
No 473 **Ulf Söderman:** On Conceptual Modelling of Mode Switching Systems, 1995.
No 475 **Choong-ho Yi:** Reasoning about Concurrent Actions in the Trajectory Semantics, 1995.
No 476 **Bo Lagerström:** Successiv resultatavräkning av pågående arbeten. - Fallstudier i tre byggföretag, 1995.
No 478 **Peter Jonsson:** Complexity of State-Variable Planning under Structural Restrictions, 1995.
FHS 7/95 **Anders Avdic:** Arbetsintegrerad systemutveckling med kalkylprogram, 1995.
No 482 **Eva L Ragnemalm:** Towards Student Modelling through Collaborative Dialogue with a Learning Companion, 1995.
No 488 **Eva Toller:** Contributions to Parallel Multiparadigm Languages: Combining Object-Oriented and Rule-Based Programming, 1995.
No 489 **Erik Stoy:** A Petri Net Based Unified Representation for Hardware/Software Co-Design, 1995.
No 497 **Johan Herber:** Environment Support for Building Structured Mathematical Models, 1995.
No 498 **Stefan Svenberg:** Structure-Driven Derivation of Inter-Lingual Functor-Argument Trees for Multi-Lingual Generation, 1995.
No 503 **Hee-Cheol Kim:** Prediction and Postdiction under Uncertainty, 1995.
FHS 8/95 **Dan Fristedt:** Metoder i användning - mot förbättring av systemutveckling genom situationell metodkunskap och metodanalys, 1995.
FHS 9/95 **Malin Bergvall:** Systemförvaltning i praktiken - en kvalitativ studie avseende centrala begrepp, aktiviteter och ansvarsroller, 1995.
No 513 **Joachim Karlsson:** Towards a Strategy for Software Requirements Selection, 1995.
No 517 **Jakob Axelsson:** Schedulability-Driven Partitioning of Heterogeneous Real-Time Systems, 1995.
No 518 **Göran Forslund:** Toward Cooperative Advice-Giving Systems: The Expert Systems Experience, 1995.
No 522 **Jörgen Andersson:** Bilder av småföretagares ekonomistyrning, 1995.
No 538 **Staffan Flodin:** Efficient Management of Object-Oriented Queries with Late Binding, 1996.
No 545 **Vadim Engelson:** An Approach to Automatic Construction of Graphical User Interfaces for Applications in Scientific Computing, 1996.
No 546 **Magnus Werner :** Multidatabase Integration using Polymorphic Queries and Views, 1996.
FiF-a 1/96 **Mikael Lind:** Affärsprocessinriktad förändringsanalys - utveckling och tillämpning av synsätt och metod, 1996.
No 549 **Jonas Hallberg:** High-Level Synthesis under Local Timing Constraints, 1996.
No 550 **Kristina Larsen:** Förutsättningar och begränsningar för arbete på distans - erfarenheter från fyra svenska företag. 1996.
No 557 **Mikael Johansson:** Quality Functions for Requirements Engineering Methods, 1996.
No 558 **Patrik Nordling:** The Simulation of Rolling Bearing Dynamics on Parallel Computers, 1996.
No 561 **Anders Ekman:** Exploration of Polygonal Environments, 1996.
No 563 **Niclas Andersson:** Compilation of Mathematical Models to Parallel Code, 1996.

No 567 **Johan Jenvald:** Simulation and Data Collection in Battle Training, 1996.
No 575 **Niclas Ohlsson:** Software Quality Engineering by Early Identification of Fault-Prone Modules, 1996.
No 576 **Mikael Ericsson:** Commenting Systems as Design Support—A Wizard-of-Oz Study, 1996.
No 587 **Jörgen Lindström:** Chefers användning av kommunikationsteknik, 1996.
No 589 **Esa Falkenroth:** Data Management in Control Applications - A Proposal Based on Active Database Systems, 1996.
No 591 **Niclas Wahllöf:** A Default Extension to Description Logics and its Applications, 1996.
No 595 **Annika Larsson:** Ekonomisk Styrning och Organisatorisk Passion - ett interaktivt perspektiv, 1997.
No 597 **Ling Lin:** A Value-based Indexing Technique for Time Sequences, 1997.
No 598 **Rego Granlund:** $C^3$Fire - A Microworld Supporting Emergency Management Training, 1997.
No 599 **Peter Ingels:** A Robust Text Processing Technique Applied to Lexical Error Recovery, 1997.
No 607 **Per-Arne Persson:** Toward a Grounded Theory for Support of Command and Control in Military Coalitions, 1997.
No 609 **Jonas S Karlsson:** A Scalable Data Structure for a Parallel Data Server, 1997.
FiF-a 4 **Carita Åbom:** Videomötesteknik i olika affärssituationer - möjligheter och hinder, 1997.
FiF-a 6 **Tommy Wedlund**: Att skapa en företagsanpassad systemutvecklingsmodell - genom rekonstruktion, värdering och vidareutveckling i T50-bolag inom ABB, 1997.
No 615 **Silvia Coradeschi**: A Decision-Mechanism for Reactive and Coordinated Agents, 1997.
No 623 **Jan Ollinen:** Det flexibla kontorets utveckling på Digital - Ett stöd för multiflex? 1997.
No 626 **David Byers:** Towards Estimating Software Testability Using Static Analysis, 1997.
No 627 **Fredrik Eklund:** Declarative Error Diagnosis of GAPLog Programs, 1997.
No 629 **Gunilla Ivefors:** Krigsspel och Informationsteknik inför en oförutsägbar framtid, 1997**.**
No 631 **Jens-Olof Lindh:** Analysing Traffic Safety from a Case-Based Reasoning Perspective, 1997
No 639 **Jukka Mäki-Turja:**. Smalltalk - a suitable Real-Time Language, 1997.
No 640 **Juha Takkinen**: CAFE: Towards a Conceptual Model for Information Management in Electronic Mail, 1997.
No 643 **Man Lin**: Formal Analysis of Reactive Rule-based Programs, 1997.
No 653 **Mats Gustafsson**: Bringing Role-Based Access Control to Distributed Systems, 1997.
FiF-a 13 **Boris Karlsson:** Metodanalys för förståelse och utveckling av systemutvecklingsverksamhet. Analys och värdering av systemutvecklingsmodeller och dess användning, 1997.
No 674 **Marcus Bjäreland:** Two Aspects of Automating Logics of Action and Change - Regression and Tractability, 1998.
No 676 **Jan Håkegård**: Hierarchical Test Architecture and Board-Level Test Controller Synthesis, 1998.
No 668 **Per-Ove Zetterlund**: Normering av svensk redovisning - En studie av tillkomsten av Redovisningsrådets rekommendation om koncernredovisning (RR01:91), 1998.
No 675 **Jimmy Tjäder**: Projektledaren & planen - en studie av projektledning i tre installations- och systemutvecklingsprojekt, 1998.
FiF-a 14 **Ulf Melin**: Informationssystem vid ökad affärs- och processorientering - egenskaper, strategier och utveckling, 1998.
No 695 **Tim Heyer**: COMPASS: Introduction of Formal Methods in Code Development and Inspection, 1998.
No 700 **Patrik Hägglund:** Programming Languages for Computer Algebra, 1998.
FiF-a 16 **Marie-Therese Christiansson:** Inter-organisatorisk verksamhetsutveckling - metoder som stöd vid utveckling av partnerskap och informationssystem, 1998.
No 712 **Christina Wennestam:** Information om immateriella resurser. Investeringar i forskning och utveckling samt i personal inom skogsindustrin, 1998.
No 719 **Joakim Gustafsson:** Extending Temporal Action Logic for Ramification and Concurrency, 1998.
No 723 **Henrik André-Jönsson:** Indexing time-series data using text indexing methods, 1999.
No 725 **Erik Larsson:** High-Level Testability Analysis and Enhancement Techniques, 1998.
No 730 **Carl-Johan Westin:** Informationsförsörjning: en fråga om ansvar - aktiviteter och uppdrag i fem stora svenska organisationers operativa informationsförsörjning, 1998.
No 731 **Åse Jansson:** Miljöhänsyn - en del i företags styrning, 1998.
No 733 **Thomas Padron-McCarthy:** Performance-Polymorphic Declarative Queries, 1998.
No 734 **Anders Bäckström:** Värdeskapande kreditgivning - Kreditriskhantering ur ett agentteoretiskt perspektiv, 1998.
FiF-a 21 **Ulf Seigerroth:** Integration av förändringsmetoder - en modell för välgrundad metodintegration, 1999.
FiF-a 22 **Fredrik Öberg:** Object-Oriented Frameworks - A New Strategy for Case Tool Development, 1998.
No 737 **Jonas Mellin:** Predictable Event Monitoring, 1998.
No 738 **Joakim Eriksson:** Specifying and Managing Rules in an Active Real-Time Database System, 1998.
FiF-a 25 **Bengt E W Andersson:** Samverkande informationssystem mellan aktörer i offentliga åtaganden - En teori om aktörsarenor i samverkan om utbyte av information, 1998.
No 742 **Pawel Pietrzak:** Static Incorrectness Diagnosis of CLP (FD), 1999.
No 748 **Tobias Ritzau:** Real-Time Reference Counting in RT-Java, 1999.
No 751 **Anders Ferntoft:** Elektronisk affärskommunikation - kontaktkostnader och kontaktprocesser mellan kunder och leverantörer på producentmarknader, 1999.
No 752 **Jo Skåmedal:** Arbete på distans och arbetsformens påverkan på resor och resmönster, 1999.
No 753 **Johan Alvehus:** Mötets metaforer. En studie av berättelser om möten, 1999.

No 754 **Magnus Lindahl:** Bankens villkor i låneavtal vid kreditgivning till högt belånade företagsförvärv: En studie ur ett agentteoretiskt perspektiv, 2000.
No 766 **Martin V. Howard:** Designing dynamic visualizations of temporal data, 1999.
No 769 **Jesper Andersson:** Towards Reactive Software Architectures, 1999.
No 775 **Anders Henriksson:** Unique kernel diagnosis, 1999.
FiF-a 30 **Pär J. Ågerfalk:** Pragmatization of Information Systems - A Theoretical and Methodological Outline, 1999.
No 787 **Charlotte Björkegren:** Learning for the next project - Bearers and barriers in knowledge transfer within an organisation, 1999.
No 788 **Håkan Nilsson:** Informationsteknik som drivkraft i granskningsprocessen - En studie av fyra revisionsbyråer, 2000.
No 790 **Erik Berglund:** Use-Oriented Documentation in Software Development, 1999.
No 791 **Klas Gäre:** Verksamhetsförändringar i samband med IS-införande, 1999.
No 800 **Anders Subotic:** Software Quality Inspection, 1999.
No 807 **Svein Bergum**: Managerial communication in telework, 2000.
No 809 **Flavius Gruian:** Energy-Aware Design of Digital Systems, 2000.
FiF-a 32 **Karin Hedström:** Kunskapsanvändning och kunskapsutveckling hos verksamhetskonsulter - Erfarenheter från ett FOU-samarbete, 2000.
No 808 **Linda Askenäs:** Affärssystemet - En studie om teknikens aktiva och passiva roll i en organisation, 2000.
No 820 **Jean Paul Meynard:** Control of industrial robots through high-level task programming, 2000.
No 823 **Lars Hult:** Publika Gränsytor - ett designexempel, 2000.
No 832 **Paul Pop:** Scheduling and Communication Synthesis for Distributed Real-Time Systems, 2000.
FiF-a 34 **Göran Hultgren:** Nätverksinriktad Förändringsanalys - perspektiv och metoder som stöd för förståelse och utveckling av affärsrelationer och informationssystem, 2000.
No 842 **Magnus Kald:** The role of management control systems in strategic business units, 2000.
No 844 **Mikael Cäker:** Vad kostar kunden? Modeller för intern redovisning, 2000.
FiF-a 37 **Ewa Braf**: Organisationers kunskapsverksamheter - en kritisk studie av ”knowledge management”, 2000.
FiF-a 40 **Henrik Lindberg:** Webbaserade affärsprocesser - Möjligheter och begränsningar, 2000.
FiF-a 41 **Benneth Christiansson:** Att komponentbasera informationssystem - Vad säger teori och praktik?, 2000.
No. 854 **Ola Pettersson:** Deliberation in a Mobile Robot, 2000.
No 863 **Dan Lawesson**: Towards Behavioral Model Fault Isolation for Object Oriented Control Systems, 2000.
No 881 **Johan Moe:** Execution Tracing of Large Distributed Systems, 2001.
No 882 **Yuxiao Zhao:** XML-based Frameworks for Internet Commerce and an Implementation of B2B e-procurement, 2001.
No 890 **Annika Flycht-Eriksson:** Domain Knowledge Management in Information-providing Dialogue systems, 2001.
FiF-a 47 **Per-Arne Segerkvis**t: Webbaserade imaginära organisationers samverkansformer: Informationssystemarkitektur och aktörssamverkan som förutsättningar för affärsprocesser, 2001.
No 894 **Stefan Svarén:** Styrning av investeringar i divisionaliserade företag - Ett koncernperspektiv, 2001.
No 906 **Lin Han**: Secure and Scalable E-Service Software Delivery, 2001.
No 917 **Emma Hansson:** Optionsprogram för anställda - en studie av svenska börsföretag, 2001.
No 916 **Susanne Odar:** IT som stöd för strategiska beslut, en studie av datorimplementerade modeller av verksamhet som stöd för beslut om anskaffning av JAS 1982, 2002.
FiF-a-49 **Stefan Holgersson:** IT-system och filtrering av verksamhetskunskap - kvalitetsproblem vid analyser och beslutsfattande som bygger på uppgifter hämtade från polisens IT-system, 2001.
FiF-a-51 **Per Oscarsson:** Informationssäkerhet i verksamheter - begrepp och modeller som stöd för förståelse av informationssäkerhet och dess hantering, 2001.
No 919 **Luis Alejandro Cortes:** A Petri Net Based Modeling and Verification Technique for Real-Time Embedded Systems, 2001.
No 915 **Niklas Sandell:** Redovisning i skuggan av en bankkris - Värdering av fastigheter. 2001.
No 931 **Fredrik Elg:** Ett dynamiskt perspektiv på individuella skillnader av heuristisk kompetens, intelligens, mentala modeller, mål och konfidens i kontroll av mikrovärlden Moro, 2002.
No 933 **Peter Aronsson:** Automatic Parallelization of Simulation Code from Equation Based Simulation Languages, 2002.
No 938 **Bourhane Kadmiry**: Fuzzy Control of Unmanned Helicopter, 2002.
No 942 **Patrik Haslum**: Prediction as a Knowledge Representation Problem: A Case Study in Model Design, 2002.
No 956 **Robert Sevenius:** On the instruments of governance - A law & economics study of capital instruments in limited liability companies, 2002.
FiF-a 58 **Johan Petersson:** Lokala elektroniska marknadsplatser - informationssystem för platsbundna affärer, 2002.
No 964 **Peter Bunus:** Debugging and Structural Analysis of Declarative Equation-Based Languages, 2002.
No 973 **Gert Jervan:** High-Level Test Generation and Built-In Self-Test Techniques for Digital Systems, 2002.
No 958 **Fredrika Berglund:** Management Control and Strategy - a Case Study of Pharmaceutical Drug Development, 2002.
FiF-a 61 **Fredrik Karlsson:** Meta-Method for Method Configuration - A Rational Unified Process Case, 2002.
No 985 **Sorin Manolache:** Schedulability Analysis of Real-Time Systems with Stochastic Task Execution Times, 2002.
No 982 **Diana Szentiványi:** Performance and Availability Trade-offs in Fault-Tolerant Middleware, 2002.
No 989 **Iakov Nakhimovski:** Modeling and Simulation of Contacting Flexible Bodies in Multibody Systems, 2002.
No 990 **Levon Saldamli:** PDEModelica - Towards a High-Level Language for Modeling with Partial Differential Equations, 2002.
No 991 **Almut Herzog:** Secure Execution Environment for Java Electronic Services, 2002.

No 999 **Jon Edvardsson:** Contributions to Program- and Specification-based Test Data Generation, 2002.
No 1000 **Anders Arpteg:** Adaptive Semi-structured Information Extraction, 2002.
No 1001 **Andrzej Bednarski:** A Dynamic Programming Approach to Optimal Retargetable Code Generation for Irregular Architectures, 2002.
No 988 **Mattias Arvola:** Good to use! : Use quality of multi-user applications in the home, 2003.
FiF-a 62 **Lennart Ljung:** Utveckling av en projektivitetsmodell - om organisationers förmåga att tillämpa projektarbetsformen, 2003.
No 1003 **Pernilla Qvarfordt:** User experience of spoken feedback in multimodal interaction, 2003.
No 1005 **Alexander Siemers:** Visualization of Dynamic Multibody Simulation With Special Reference to Contacts, 2003.
No 1008 **Jens Gustavsson:** Towards Unanticipated Runtime Software Evolution, 2003.
No 1010 **Calin Curescu:** Adaptive QoS-aware Resource Allocation for Wireless Networks, 2003.
No 1015 **Anna Andersson:** Management Information Systems in Process-oriented Healthcare Organisations, 2003.
No 1018 **Björn Johansson:** Feedforward Control in Dynamic Situations, 2003.
No 1022 **Traian Pop:** Scheduling and Optimisation of Heterogeneous Time/Event-Triggered Distributed Embedded Systems, 2003.
FiF-a 65 **Britt-Marie Johansson:** Kundkommunikation på distans - en studie om kommunikationsmediets betydelse i affärstransaktioner, 2003.
No 1024 **Aleksandra Tešanovic:** Towards Aspectual Component-Based Real-Time System Development, 2003.
No 1034 **Arja Vainio-Larsson**: Designing for Use in a Future Context - Five Case Studies in Retrospect, 2003.
No 1033 **Peter Nilsson:** Svenska bankers redovisningsval vid reservering för befarade kreditförluster - En studie vid införandet av nya redovisningsregler, 2003.
FiF-a 69 **Fredrik Ericsson:** Information Technology for Learning and Acquiring of Work Knowledge, 2003.
No 1049 **Marcus Comstedt:** Towards Fine-Grained Binary Composition through Link Time Weaving, 2003.
No 1052 **Åsa Hedenskog:** Increasing the Automation of Radio Network Control, 2003.
No 1054 **Claudiu Duma:** Security and Efficiency Tradeoffs in Multicast Group Key Management, 2003.
FiF-a 71 **Emma Eliason:** Effektanalys av IT-systems handlingsutrymme, 2003.
No 1055 **Carl Cederberg:** Experiments in Indirect Fault Injection with Open Source and Industrial Software, 2003.
No 1058 **Daniel Karlsson:** Towards Formal Verification in a Component-based Reuse Methodology, 2003.
FiF-a 73 **Anders Hjalmarsson:** Att etablera och vidmakthålla förbättringsverksamhet - behovet av koordination och interaktion vid förändring av systemutvecklingsverksamheter, 2004.
No 1079 **Pontus Johansson:** Design and Development of Recommender Dialogue Systems, 2004.
No 1084 **Charlotte Stoltz:** Calling for Call Centres - A Study of Call Centre Locations in a Swedish Rural Region, 2004.
FiF-a 74 **Björn Johansson:** Deciding on Using Application Service Provision in SMEs, 2004.
No 1094 **Genevieve Gorrell:** Language Modelling and Error Handling in Spoken Dialogue Systems, 2004.
No 1095 **Ulf Johansson:** Rule Extraction - the Key to Accurate and Comprehensible Data Mining Models, 2004.
No 1099 **Sonia Sangari:** Computational Models of Some Communicative Head Movements, 2004.
No 1110 **Hans Nässla:** Intra-Family Information Flow and Prospects for Communication Systems, 2004.
No 1116 **Henrik Sällberg:** On the value of customer loyalty programs - A study of point programs and switching costs, 2004.
FiF-a 77 **Ulf Larsson:** Designarbete i dialog - karaktärisering av interaktionen mellan användare och utvecklare i en systemutvecklingsprocess, 2004.
No 1126 **Andreas Borg:** Contribution to Management and Validation of Non-Functional Requirements, 2004.
No 1127 **Per-Ola Kristensson:** Large Vocabulary Shorthand Writing on Stylus Keyboard, 2004.
No 1132 **Pär-Anders Albinsson:** Interacting with Command and Control Systems: Tools for Operators and Designers, 2004.
No 1130 **Ioan Chisalita:** Safety-Oriented Communication in Mobile Networks for Vehicles, 2004.
No 1138 **Thomas Gustafsson:** Maintaining Data Consistency in Embedded Databases for Vehicular Systems, 2004.
No 1149 **Vaida Jakonienė:** A Study in Integrating Multiple Biological Data Sources, 2005.
No 1156 **Abdil Rashid Mohamed**: High-Level Techniques for Built-In Self-Test Resources Optimization, 2005.
No 1162 **Adrian Pop:** Contributions to Meta-Modeling Tools and Methods, 2005.
No 1165 **Fidel Vascós Palacios:** On the information exchange between physicians and social insurance officers in the sick leave process: an Activity Theoretical perspective, 2005.
FiF-a 84 **Jenny Lagsten:** Verksamhetsutvecklande utvärdering i informationssystemprojekt, 2005.
No 1166 **Emma Larsdotter Nilsson:** Modeling, Simulation, and Visualization of Metabolic Pathways Using Modelica, 2005.
No 1167 **Christina Keller:** Virtual Learning Environments in higher education. A study of students' acceptance of educational technology, 2005.
No 1168 **Cécile Åberg:** Integration of organizational workflows and the Semantic Web, 2005.
FiF-a 85 **Anders Forsman:** Standardisering som grund för informationssamverkan och IT-tjänster - En fallstudie baserad på trafikinformationstjänsten RDS-TMC, 2005.
No 1171 **Yu-Hsing Huang:** A systemic traffic accident model, 2005.
FiF-a 86 **Jan Olausson:** Att modellera uppdrag - grunder för förståelse av processinriktade informationssystem i transaktionsintensiva verksamheter, 2005.
No 1172 **Petter Ahlström**: Affärsstrategier för seniorbostadsmarknaden, 2005.
No 1183 **Mathias Cöster**: Beyond IT and Productivity - How Digitization Transformed the Graphic Industry, 2005.
No 1184 **Åsa Horzella**: Beyond IT and Productivity - Effects of Digitized Information Flows in Grocery Distribution, 2005.
No 1185 **Maria Kollberg**: Beyond IT and Productivity - Effects of Digitized Information Flows in the Logging Industry, 2005.
No 1190 **David Dinka**: Role and Identity - Experience of technology in professional settings, 2005.

No 1191 **Andreas Hansson**: Increasing the Storage Capacity of Recursive Auto-associative Memory by Segmenting Data, 2005.
No 1192 **Nicklas Bergfeldt:** Towards Detached Communication for Robot Cooperation, 2005.
No 1194 **Dennis Maciuszek:** Towards Dependable Virtual Companions for Later Life, 2005.
No 1204 **Beatrice Alenljung:** Decision-making in the Requirements Engineering Process: A Human-centered Approach, 2005.
No 1206 **Anders Larsson:** System-on-Chip Test Scheduling and Test Infrastructure Design, 2005.
No 1207 **John Wilander:** Policy and Implementation Assurance for Software Security, 2005.
No 1209 **Andreas Käll:** Översättningar av en managementmodell - En studie av införandet av Balanced Scorecard i ett landsting, 2005.
No 1225 **He Tan:** Aligning and Merging Biomedical Ontologies, 2006.
No 1228 **Artur Wilk:** Descriptive Types for XML Query Language Xcerpt, 2006.
No 1229 **Per Olof Pettersson:** Sampling-based Path Planning for an Autonomous Helicopter, 2006.
No 1231 **Kalle Burbeck:** Adaptive Real-time Anomaly Detection for Safeguarding Critical Networks, 2006.
No 1233 **Daniela Mihailescu:** Implementation Methodology in Action: A Study of an Enterprise Systems Implementation Methodology, 2006.
No 1244 **Jörgen Skågeby:** Public and Non-public gifting on the Internet, 2006.
No 1248 **Karolina Eliasson:** The Use of Case-Based Reasoning in a Human-Robot Dialog System, 2006.
No 1263 **Misook Park-Westman:** Managing Competence Development Programs in a Cross-Cultural Organisation - What are the Barriers and Enablers, 2006.
FiF-a 90 **Amra Halilovic:** Ett praktikperspektiv på hantering av mjukvarukomponenter, 2006.
No 1272 **Raquel Flodström**: A Framework for the Strategic Management of Information Technology, 2006.
No 1277 **Viacheslav Izosimov:** Scheduling and Optimization of Fault-Tolerant Embedded Systems, 2006.
No 1283 **Håkan Hasewinkel:** A Blueprint for Using Commercial Games off the Shelf in Defence Training, Education and Research Simulations, 2006.
FiF-a 91 **Hanna Broberg:** Verksamhetsanpassade IT-stöd - Designteori och metod, 2006.
No 1286 **Robert Kaminski:** Towards an XML Document Restructuring Framework, 2006.
No 1293 **Jiri Trnka:** Prerequisites for data sharing in emergency management, 2007.
No 1302 **Björn Hägglund:** A Framework for Designing Constraint Stores, 2007.
No 1303 **Daniel Andreasson**: Slack-Time Aware Dynamic Routing Schemes for On-Chip Networks, 2007.
No 1305 **Magnus Ingmarsson:** Modelling User Tasks and Intentions for Service Discovery in Ubiquitous Computing, 2007.
No 1306 **Gustaf Svedjemo**: Ontology as Conceptual Schema when Modelling Historical Maps for Database Storage, 2007.
No 1307 **Gianpaolo Conte**: Navigation Functionalities for an Autonomous UAV Helicopter, 2007.
No 1309 **Ola Leifler**: User-Centric Critiquing in Command and Control: The DKExpert and ComPlan Approaches, 2007.
No 1312 **Henrik Svensson**: Embodied simulation as off-line representation, 2007.
No 1313 **Zhiyuan He:** System-on-Chip Test Scheduling with Defect-Probability and Temperature Considerations, 2007.
No 1317 **Jonas Elmqvist:** Components, Safety Interfaces and Compositional Analysis, 2007.
No 1320 **Håkan Sundblad:** Question Classification in Question Answering Systems, 2007.
No 1323 **Magnus Lundqvist**: Information Demand and Use: Improving Information Flow within Small-scale Business Contexts, 2007.
No 1329 **Martin Magnusson:** Deductive Planning and Composite Actions in Temporal Action Logic, 2007.
No 1331 **Mikael Asplund:** Restoring Consistency after Network Partitions, 2007.
No 1332 **Martin Fransson:** Towards Individualized Drug Dosage - General Methods and Case Studies, 2007.
No 1333 **Karin Camara:** A Visual Query Language Served by a Multi-sensor Environment, 2007.
No 1337 **David Broman:** Safety, Security, and Semantic Aspects of Equation-Based Object-Oriented Languages and Environments, 2007.
No 1339 **Mikhail Chalabine:** Invasive Interactive Parallelization, 2007.
No 1351 **Susanna Nilsson:** A Holistic Approach to Usability Evaluations of Mixed Reality Systems, 2008.
No 1353 **Shanai Ardi:** A Model and Implementation of a Security Plug-in for the Software Life Cycle, 2008.
No 1356 **Erik Kuiper:** Mobility and Routing in a Delay-tolerant Network of Unmanned Aerial Vehicles, 2008.
No 1359 **Jana Rambusch**: Situated Play, 2008.
No 1361 **Martin Karresand:** Completing the Picture - Fragments and Back Again, 2008.
No 1363 **Per Nyblom:** Dynamic Abstraction for Interleaved Task Planning and Execution, 2008.
No 1371 **Fredrik Lantz:** Terrain Object Recognition and Context Fusion for Decision Support, 2008.
No 1373 **Martin Östlund:** Assistance Plus: 3D-mediated Advice-giving on Pharmaceutical Products, 2008.
No 1381 **Håkan Lundvall:** Automatic Parallelization using Pipelining for Equation-Based Simulation Languages, 2008.
No 1386 **Mirko Thorstensson:** Using Observers for Model Based Data Collection in Distributed Tactical Operations, 2008.
No 1387 **Bahlol Rahimi:** Implementation of Health Information Systems, 2008.
No 1392 **Maria Holmqvist:** Word Alignment by Re-using Parallel Phrases, 2008.
No 1393 **Mattias Eriksson**: Integrated Software Pipelining, 2009.
No 1401 **Annika Öhgren:** Towards an Ontology Development Methodology for Small and Medium-sized Enterprises, 2009.
No 1410 **Rickard Holsmark:** Deadlock Free Routing in Mesh Networks on Chip with Regions, 2009.
No 1421 **Sara Stymne:** Compound Processing for Phrase-Based Statistical Machine Translation, 2009.
No 1427 **Tommy Ellqvist**: Supporting Scientific Collaboration through Workflows and Provenance, 2009.
No 1450 **Fabian Segelström:** Visualisations in Service Design, 2010.
No 1459 **Min Bao**: System Level Techniques for Temperature-Aware Energy Optimization, 2010.
No 1466 **Mohammad Saifullah**: Exploring Biologically Inspired Interactive Networks for Object Recognition, 2011

No 1468 **Qiang Liu**: Dealing with Missing Mappings and Structure in a Network of Ontologies, 2011.
No 1469 **Ruxandra Pop**: Mapping Concurrent Applications to Multiprocessor Systems with Multithreaded Processors and Network on Chip-Based Interconnections, 2011.
No 1476 **Per-Magnus Olsson**: Positioning Algorithms for Surveillance Using Unmanned Aerial Vehicles, 2011.
No 1481 **Anna Vapen**: Contributions to Web Authentication for Untrusted Computers, 2011.
No 1485 **Loove Broms:** Sustainable Interactions: Studies in the Design of Energy Awareness Artefacts, 2011.
FiF-a 101 **Johan Blomkvist:** Conceptualising Prototypes in Service Design, 2011.
No 1490 **Håkan Warnquist:** Computer-Assisted Troubleshooting for Efficient Off-board Diagnosis, 2011.
No 1503 **Jakob Rosén:** Predictable Real-Time Applications on Multiprocessor Systems-on-Chip, 2011.
No 1504 **Usman Dastgeer:** Skeleton Programming for Heterogeneous GPU-based Systems, 2011.
No 1506 **David Landén:** Complex Task Allocation for Delegation: From Theory to Practice, 2011.
No 1507 **Kristian Stavåker**: Contributions to Parallel Simulation of Equation-Based Models on Graphics Processing Units, 2011.
No 1509 **Mariusz Wzorek:** Selected Aspects of Navigation and Path Planning in Unmanned Aircraft Systems, 2011.
No 1510 **Piotr Rudol:** Increasing Autonomy of Unmanned Aircraft Systems Through the Use of Imaging Sensors, 2011.
No 1513 **Anders Carstensen:** The Evolution of the Connector View Concept: Enterprise Models for Interoperability Solutions in the Extended Enterprise, 2011.
No 1523 **Jody Foo:** Computational Terminology: Exploring Bilingual and Monolingual Term Extraction, 2012.
No 1550 **Anders Fröberg:** Models and Tools for Distributed User Interface Development, 2012.
No 1558 **Dimitar Nikolov:** Optimizing Fault Tolerance for Real-Time Systems, 2012.
No 1582 **Dennis Andersson:** Mission Experience: How to Model and Capture it to Enable Vicarious Learning, 2013.
No 1586 **Massimiliano Raciti:** Anomaly Detection and its Adaptation: Studies on Cyber-physical Systems, 2013.
No 1588 **Banafsheh Khademhosseinieh:** Towards an Approach for Efficiency Evaluation of Enterprise Modeling Methods, 2013.
No 1589 **Amy Rankin:** Resilience in High Risk Work: Analysing Adaptive Performance, 2013.
No 1592 **Martin Sjölund:** Tools for Understanding, Debugging, and Simulation Performance Improvement of Equation-Based Models, 2013.
No 1606 **Karl Hammar:** Towards an Ontology Design Pattern Quality Model, 2013.
No 1624 **Maria Vasilevskaya:** Designing Security-enhanced Embedded Systems: Bridging Two Islands of Expertise, 2013.
No 1627 **Ekhiotz Vergara:** Exploiting Energy Awareness in Mobile Communication, 2013.
No 1644 **Valentina Ivanova:** Integration of Ontology Alignment and Ontology Debugging for Taxonomy Networks, 2014.
No 1647 **Dag Sonntag:** A Study of Chain Graph Interpretations, 2014.