Linköping Studies in Science and Technology

Dissertations. No. 1748

Chain Graphs
– Interpretations, Expressiveness and Learning Algorithms

by

Dag Sonntag

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

Linköping 2016

ISBN 978-91-7685-818-9
ISSN 0345–7524

Cover by Dag Sonntag
Original art by Emil Widlund

Printed by LiU Tryck 2016

URL: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-125921

Abstract

Probabilistic graphical models are currently one of the most commonly used architectures for modelling and reasoning with uncertainty. The most widely used subclass of these models is directed acyclic graphs, also known as Bayesian networks, which are used in a wide range of applications both in research and industry. Directed acyclic graphs do, however, have a major limitation, which is that only asymmetric relationships, namely cause and effect relationships, can be modelled between their variables. A class of probabilistic graphical models that tries to address this shortcoming is chain graphs, which include two types of edges in the models representing both symmetric and asymmetric relationships between the variables. This allows for a wider range of independence models to be modelled and depending on how the second edge is interpreted, we also have different so-called chain graph interpretations.

Although chain graphs were first introduced in the late eighties, most research on probabilistic graphical models naturally started in the least complex subclasses, such as directed acyclic graphs and undirected graphs. The field of chain graphs has therefore been relatively dormant. However, due to the maturity of the research field of probabilistic graphical models and the rise of more data-driven approaches to system modelling, chain graphs have recently received renewed interest in research. In this thesis we provide an introduction to chain graphs where we incorporate the progress made in the field. More specifically, we study the three chain graph interpretations that exist in research in terms of their separation criteria, their possible parametrizations and the intuition behind their edges. In addition to this we also compare the expressivity of the interpretations in terms of representable independence models as well as propose new structure learning algorithms to learn chain graph models from data.

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

Populärvetenskaplig sammanfattning

Inom statistik, fysik och datavetenskap har modeller använts genom historien för att förstå och beskriva olika delar av världen. I de system man försökt beskriva har då de relevanta faktorerna ofta representerats som variabler och relationerna mellan dessa variabler har på olika sätt avspeglats i modellerna. Beroende på karakteristiken av systemet, och dess variabler, har olika typer av modeller utvecklats och använts men en vanlig del har varit att använda någon typ av grafisk illustration för att underlätta förståelsen av hur variablerna relaterar till varandra. I denna avhandling behandlar vi en typ av sådan modell kallad sannolikhetsbaserad grafisk modell (probabilistic graphical model) och mer specifikt en underklass av dessa kallade kedje-grafer (chain graphs). I en sannolikhetsbaserad grafisk modell representeras variablerna som noder och relationerna mellan variablerna som olika typer av bågar mellan dessa noder. Variablerna kan vara av olika natur, men oftast är de antingen diskreta, alltså att de kan vara i ett av flera tillstånd, eller kontinuerliga, d.v.s. att de kan ta ett värde på en kontinuerlig skala. Till grafen kan det också finnas en parametrisering av relationerna mellan variablerna som i sig är sannolikhetsbaserad och där en variabel endast påverkas av närliggande variabler i grafen. Sannolikhetsbaserad betyder för t.ex. en diskret variabel att dess exakta tillstånd inte behöver vara känt, utan att man istället uppskattar sannolikheterna att variabeln befinner sig i sina olika tillstånd. Observerar man sedan att en viss variabel befinner sig i ett visst tillstånd kan man beräkna hur detta påverkar sannolikheterna av att närliggande variabler befinner sig i sina respektive tillstånd och hur detta i sin tur påverkar de andra variablerna i modellen. Att använda en sannolikhetsbaserad modell, och inte en deterministisk modell, medför många fördelar. Det tillåter t.ex. att modellen kan inkludera relationer mellan faktorer där den exakta påverkan mellan dem är svår eller omöjlig att bestämma. Man kan även utelämna faktorer helt ur sin modell, antingen för att de är okända eller svåra att mäta, och den okända påverkan dessa faktorer har på andra variabler i modellen inkorporeras automatiskt som osäkerhet i tillstånden av variablerna. Sannolikhetsbaseringen gör det också enkelt att uppskatta parametrarna för modellen av antingen en expert av

systemet eller från insamlad data. Detta, tillsammans med grafen som ger en intuitiv översikt av modellen och dess representerade relationer, har gjort att sannolikhetsbaserade grafiska modeller idag är en av de mest välanvända modelltyper som hanterar icke-deterministiska system. Olika underklasser av dessa modeller används idag i en uppsjö av applikationer som t.ex. automatisk felsökning av skrivare, modellering av protein-strukturer i bioinformatik eller beslutsstödssystem inom finansmarknadsanalys. Ett problem med de underklasser av sannolikhetsbaserade grafiska modeller som används mest idag är dock att de endast kan representera en typ av relation mellan sina variabler. För den mest välanvända underklassen, Bayesianska nätverk, är detta den asymmetriska relationen orsak-och-påverkan, där en variabel är orsaken av tillståndet av en annan variabel. Det finns dock system som innehåller andra typer av relationer, som t.ex. när det finns okända gemensamma orsaksfaktorer för flera variabler i systemet. Bayesianska nätverk kan då visserligen fortfarande representera systemet, men bara genom att på ett inkorrekt sätt lägga till extra bågar i modellen och därmed representera inkorrekta relationer. Detta är problematiskt ur bland annat följande aspekter.

- Modellen kan skilja sig från en experts förståelse av systemet vilket gör modellen svår att förstå och parametrisera.
- Aktiv påverkan av systemet kan ge ett annorlunda resultat i modellen jämfört med i verkligheten.
- De algoritmer som används för att beräkna sannolikheten för tillstånden av variablerna blir långsammare ju fler bågar grafen innehåller.

För att lösa detta problem har flera olika metoder föreslagits. I denna avhandling diskuterar vi en sådan metod som går ut på att använda en mer expressiv underklass av sannolikhetsbaserade grafiska modeller än Bayesianska nätverk, d.v.s. en underklass som kan representera en större mängd system. Underklassen som avhandlas kallas kedje-grafer och som förutom relationen orsak-och-påverkan också kan representera en symmetrisk relation. Denna symmetriska relation kan dock tolkas på olika sätt, vilket har gett upphov till flera så kallade interpretationer. I avhandlingen undersöker vi närmare de tre interpretationerna som fått mest uppmärksamhet i forskningen; Lauritzen-Wermuth-Frydenberg interpretationen (LWF), Andersson-Madigan-Perlman interpretationen (AMP) samt multivariata regressionsinterpretationen (MVR). Mer specifikt beskriver vi, för var och en av interpretationerna, hur de fungerar, hur bågarna i graferna kan tolkas samt vilka parametriseringar som finns. Vi undersöker också hur många olika system de olika interpretationerna kan representera samt presenterar algoritmer för att lära sig kedje-grafsmodeller från existerande data.

Acknowledgements

As I am writing this I am closing in on the end of my five year journey as a PhD student here at Linköpings University, and what a journey it has been. During these years I have expanded my understanding of the world and learnt to push my abstract reasoning to levels I had not thought possible before. However, I can without a doubt say that this journey would not have been possible without the help of the people around me.

Hence I would like to thank my colleges here at IDA who have made this workplace a joy to work at. I can of course not mention you all, but some of you deserve extra recognition.

Of my close colleagues, I would first like to thank Nahid Shahmehri, head of my division and my secondary supervisor, for pushing me to become a better researcher and to be on the top of my game. Without you this division would not be nearly as good as it is today. Marcus Bendtsen, my PhD brother and travel companion, thank you for your good company and for being my sounding board when I have been stuck. You have really helped me understand the world better by introducing me to the world of economics. I would also like to thank Vengatanathan Krishnamoorthi, Zlatan Dragisic, Valentina Ivanova and Patrick Lambrix for your strange but interesting lunch discussions over the years.

From the IDA administrative personnel, I would especially like to thank Karin Hendry and Anne Moe for your tireless help and patience. How you must get tired of us PhD students and our endless questions. In addition to this, I would like to thank Brittany Shahmehri for proofreading this thesis. Your feedback has allowed me to improve the language considerably.

However, it is not only people at IDA that have helped me on my journey. In the European probabilistic graphical model research community I was welcomed with open arms and I would like to thank the researchers there for your valuable feedback over the years. Then there are of course my family and friends who have supported me, both by encouragement and by offering distractions when things have been tough. Thank you for that!

Last, but definitely not least, I would like to thank my PhD supervisor Jose M. Peña. Your importance for my success cannot be overstated and you have truly been the best supervisor anyone could have wished for. I would like to thank you from the bottom of my heart for all your encouragement,

help and patience. I wish you the best of success in your future endeavours and you will always be a role model of mine.

So thank you all! Now new adventures await and I hope I will have the pleasure of working with you again in the not too far future!

Dag Sonntag
March 2016
Linköping, Sweden

Contents

Chapter 1

Introduction

In fields like physics, statistics and computer science models have been used throughout history to understand and describe the relationships between certain aspects in the world. In the models the relevant aspects have typically been represented as variables, forming a system, and depending on the nature of the aspects and relationships different types of models have been used. In this thesis we will look into one such type of model called probabilistic graphical model (PGM), and more specifically its subclass, chain graphs. PGMs are based on the idea that the relationships between the variables in a system are inherently uncertain (probabilistic) and can be described according to a graph. The uncertainty can be due to a number of factors, the most important being that only parts of the system might be observable and that measurements might be noisy, whereas representing the model as a graph allows for an intuitive representation of the interactions between the variables in the system. The graph also allows us to reason how information may propagate throughout our model when for example the states of some variables in the system are observed. PGMs were introduced at the beginning of the last century with Wrights' path analysis [39] and Gibbs' applications to statistical physics [10], but did not receive much attention until the 1980s when Pearl applied the models in computer science. After that the research field bloomed and PGMs are currently used in a wide range of applications, for example error diagnostics in printers, modelling protein structures in bioinformatics and decision support systems in market analysis. The main advantages of using PGMs compared to other models are that the representation is intuitive, inference can be done efficiently and a wide range of algorithms exist for creating the models from data. As a result, PGMs have arguably become the most important architecture for reasoning with uncertainty [17].

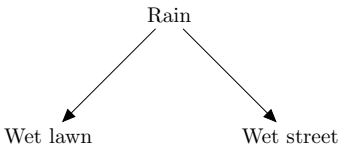
To model a system as a PGM, we first need to identify the variables that are of interest. Depending on the nature of the variables they can be modelled differently. The most researched cases are when the variables are

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 continuous range. The graph of a PGM then represents these variables as nodes and the relationships between the variables as edges where the type of relationship represented differs depending on which PGM subclass is used. In addition to the graph a PGM class can also contain some parametrization of the relationships between the variables in the model based on the graph. The parameters define the probability that a variable takes a certain state or value depending on the state or values of its neighboring variables in the graph, and are typically represented as tables for discrete variables and as functions for continuous variables. Hence we can say that the graph of a PGM represents which variables interact in the modelled system, while the parametrization represents how they interact. An example of a PGM is shown in Figure 1.1 which will be explained in detail later.

One of the most basic subclasses of PGMs is undirected graphs (UGs), also known as Markov networks, in which each undirected edge represents a direct correlation between the two variables it connects, while no edge means that the variables are not directly correlated. The most well known and widely used PGM class, however, is directed acyclic graphs (DAGs), also known as Bayesian networks.[1] In a DAG the directed edges can be seen as representing cause and effect relationships. As an example, we can consider a system with the following 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 DAG would result in the graph shown in Figure 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 0.05 otherwise. 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 Figure 1.1b.

Using this DAG 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 *On 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?*. From the causal assumptions, such as that the rain causes the lawn to be wet, it also follows which variables may be dependent on which other variables, i.e. when observations about certain variables may affect the probabilities of other variables taking certain states or values. We can, for example, deduce that observing whether the lawn is wet may change our belief about whether the street is wet if we have not observed whether it has

[1] In this thesis we make no distinction between DAGs and Bayesian networks.



(a) The graph 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.

	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

(c) A joint probability distribution.

Figure 1.1: A simple DAG.

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. On the other hand, if we have observed that it has been raining, then observing that the lawn is wet does not affect our belief about whether the street is wet or not. This is because observing that the street is wet does not change our belief about whether it has been raining or not, since we already know this. Hence we say that the wet street variable is independent of the wet lawn variable given the rain variable. These independences are encoded in the graph of a PGM according to the separation criteria of that PGM class. In the following chapters we will cover how these independences can be read from the graph, but the important thing here is that we can, from just studying the graph, conclude which variables may be dependent and which are independent of other variables given a third set of variables.

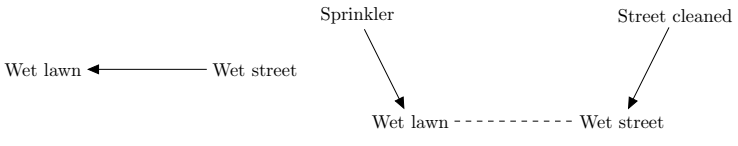
A PGM can also be seen as a factorization of a joint probability distribution of the state of a system. If we look at the example shown in Figure 1.1 we can see that the DAG represents the factorization $p(Rain, WetStreet, WetLawn)$ = $p(WetStreet|Rain)p(WetLawn|Rain)p(Rain)$ due to the independences represented in the graph. Factorizing a large joint probability distribution has many benefits. First, it illuminates the conditional independences between the variables in the distribution. Secondly, instead of having one large joint

probability distribution 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-linearly if most variables are conditionally independent. Multiple small probability distributions generally also allow us to do calculations faster than using a single joint probability distribution. An example of this is seen in Figure 1.1 where the joint probability distribution in Figure 1.1c can be factorized according to the DAG shown in Figure 1.1a into the conditional probability distributions in Figure 1.1b.

As noted above, DAGs work fine and are used in a wide range of applications today. DAGs do, however, have some shortcomings due to the fact that they only model asymmetric causal relationships between variables, i.e. relationships where the order of the variables matters. This means that when we want to model a system with some other kind of relationship between its variables, such as a symmetric relationship where the variables in the relationship have no internal ordering, the representation falls short. For example, this can happen when we have the system described for Figure 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 variables are correlated, i.e. when we observe that the lawn is wet this increases our belief that the street is also 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. That the relation is non-causal can also be seen 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.

The correct way to model such a system would instead be to use a class of models that can model the symmetric relationship between the variables, such as UGs or bidirected graphs (BGs). However, what happens if we have both symmetric and asymmetric relationships in a system? For example, consider the case where we extend the system described above with two additional variables, a sprinkler 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 we have two causal relationships ($Sprinkler \rightarrow WetLawn$, $StreetCleaned \rightarrow WetStreet$) and one non-causal symmetric relationship ($WetLawn - WetStreet$) resulting in a system that cannot be represented correctly as a DAG, UG or BG. A model including the relations described in the extended system is shown in Figure 1.2b where the unrepresentable relation is shown as a dashed line.

It can be argued that the system described above is causal and hence can be modelled as a DAG but that the model is simply missing a vari-



(a) Representable independence model.

(b) Extended unrepresentable independence model.

Figure 1.2: Non-causal relationships.

able. This variable can then be added to the model as a so-called latent, or hidden, variable that represents the unmeasured rain variable. This approach is a research field in itself and will not be covered in this thesis, but it should be clear that adding latent nodes to a model is no trivial task [27]. Moreover, there are many types of symmetric relations for which this is not possible, such as if we have a selection bias relation between two variables in our system. This means that the two variables in the relation influence whether or not the system is sampled, causing the sampling process to not be properly randomized. Another case of a symmetric relation is when two variables are tied through an equation, such as in Boyle's law $Pressure \cdot Temperature = constant$. We then know that if the pressure decreases the temperature must increase and vice versa. Hence the two variables are correlated but neither is the cause of the other and this relation is not due to some unknown factor [5].

However, currently systems containing both causal and non-causal relationships are mostly modelled with DAGs. This poses many problems, primarily that the models are incorrect with respect to the dynamics of the underlying system. This makes them hard to understand by experts in the domain and conclusions drawn from the models may be wrong. A better approach would be to use a more expressive PGM class, that can model both symmetric and asymmetric relationships, and in this thesis we describe one such PGM class called chain graphs (CGs). CGs contain two types of edges, the directed edge that corresponds to the causal relationship in DAGs and a second type of edge representing a symmetric relationship. This allows CGs to correctly model a much larger set of systems than DAGs [32] 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 there has recently been renewed interest in them as researchers have begun modelling more advanced systems, such as gene networks [2] or financial networks [6].

While the interpretation of the directed edge in a CG is quite clear, the second type of edge can represent different types of relations and, depending on how we interpret it in the graph, we say that we have different CG interpretations. Today there are several possible interpretations of

CGs with different separation criteria, i.e. different ways of reading conditional independences from the graph, and different intuitive meaning behind their edges. The first interpretation (LWF) was introduced by Lauritzen, Wermuth and Frydenberg [9, 14] to combine DAGs and UGs. The second interpretation (AMP), was introduced by Andersson, Madigan and Perlman, and also combines DAGs and UGs but with a separation criterion that more closely resembles the one of DAGs [1]. The third interpretation, the multivariate regression interpretation (MVR), was introduced by Cox and Wermuth [3] to combine DAGs and BGs. In addition to these, a fourth interpretation has been proposed [7], but the three interpretations above have received the most attention in the literature and are the ones discussed in this thesis. It is noteworthy that for LWF, AMP and MVR CGs no interpretation subsumes another [7, 30], and no interpretation is generally better than any other. LWF, AMP and MVR CG interpretations are just different from each other, similarly to the way that DAGs and UGs are different from each other, and are suited to different problems. Also, although they all have different properties, CGs are characterized 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. This means that CGs allow for a partial causal ordering where the variables within each chain component have no internal ordering, while the chain components themselves are partially ordered.

The rest of the thesis is organized as follows. It starts with seven chapters that are intended to give the reader an introduction to the research field of CGs. These are then followed by an appendix that contains the papers with our contributions to the field, where the areas discussed in previous chapters are revisited and covered in detail. The introductory chapters start with Chapter 2, where we define the notation used throughout the thesis, where after we in Chapter 3 discuss how CGs relate to other PGM classes and why they are interesting to research. This is followed by Chapter 4 where we take a closer look at each of the interpretations in terms of separation criterion, possible parametrizations and intuitive meaning of the edges. In Chapter 5 we then compare the expressivity of the different interpretations, i.e. the number of independence models, or systems, representable. Subsequently, in Chapter 6, we give an overview of what structure learning algorithms currently exist for CGs and how they perform relative to each other. Finally in Chapter 7 we conclude the introductory chapters of the thesis and discuss challenges for future research. This is then followed by Appendix A starting with a short summary, describing how the papers are related to the previous chapters, followed by the papers themselves.

Chapter 2

Notation

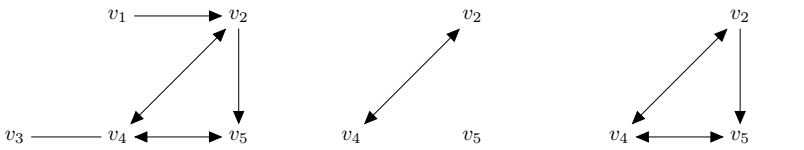
In the last chapter we discussed why PGMs and CGs are useful without going into the technical details. In the rest of the thesis we will have a more technical standpoint, hence we need to define the terms used. Note that the definitions and examples focus on the subject of the thesis, CGs, and for a more complete introduction to PGMs we refer the reader to the work by Koller and Friedman [11].

Throughout the thesis we use lower case letters to denote variables, or nodes, while for sets of variables (including singletons) we use upper-case letters. All graphs and probability distributions are defined over a finite set of variables V represented as nodes in the graph.

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

The *parents* of a set of nodes X of G is the set $pa_G(X) = \{v_i | v_i \to v_j$ is in G, $v_i \notin X$ and $v_j \in X\}$. The *children* of X is the set $ch_G(X) = \{v_i | v_j \to v_i$ is in G, $v_i \notin X$ and $v_j \in X\}$. The *spouses* of X is the set $sp_G(X) = \{v_i | v_i \leftrightarrow v_j$ is in G, $v_i \notin X$ and $v_j \in X\}$. The *neighbors* of X is the set $nb_G(X) = \{v_i | v_i - v_j$ is in G, $v_i \notin X$ and $v_j \in X\}$. 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 $ad_G(X) = bd_G(X) \cup ch_G(X)$.

To exemplify these concepts we can study the graph G with five nodes shown in Figure 2.1a. In the graph we can see two bidirected edges, one between v_2 and v_4 and one between v_4 and v_5. Hence we know the spouses of v_4 are v_2 and v_5. G also contains two directed edges from v_1 to v_2 and from v_2 to v_5 and we can see that v_5 is the only child of v_2 and v_2 is the only child of v_1. Finally G also contains one undirected edge between v_3 and v_4 and hence v_3 is a neighbor of v_4. All in all this means that the boundary of v_2 is v_1 and v_4 while the adjacents of v_2 also contains v_5 in addition to v_1 and v_4.



(a) A graph G.

(b) A subgraph of G over $\{v_2, v_4, v_5\}$.

(c) A subgraph of G induced by $\{v_2, v_4, v_5\}$.

Figure 2.1: Three different graphs.

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. A set of nodes is *complete* in a graph G if all nodes in the set are adjacent to each other in G. A complete set of nodes is also said to be a *clique* in G if there exists no superset of it that is complete.

To exemplify these concepts we can study the graph G in Figure 2.1 again. In the graph we can see that v_5 is a strict descendant of v_1 due to the strictly descending path $v_1 \to v_2 \to v_5$, while v_4 is not. v_4 is, however, in the descendants of v_1 together with v_2, v_3 and v_5. v_1 is therefore an ancestor of all variables except itself. We can also see that G contains a semi-directed cycle $v_2 \to v_5 \leftrightarrow v_4 \leftrightarrow v_2$. In Figure 2.1b we can see a subgraph of G with the variables v_2, v_4 and v_5 while in Figure 2.1c we see the subgraph of G induced by the same variables. Since the nodes v_2, v_4 and v_5 are all adjacent we can also note that this set of nodes is complete in G.

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

(a) An LWF or AMP CG H. (b) An MVR CG F.

Figure 2.2: Two different CGs.

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 an LWF CG or an AMP CG (resp. MVR CG) is a maximal set of nodes such that there is a path between every pair of nodes in C containing only undirected edges (resp. bidirected edges). An *ancestral graph* (AG) contains bidirected, undirected and directed edges but no subgraphs of the form $v_i \multimap v_j - v_k$ nor any pair of nodes v_i and v_j such that $v_j \in sde(v_i)$ and $v_i \in sp_G(v_j) \cup ch_G(v_j)$. Similarly *regression CGs* and *marginal AMP CGs* (MAMP CGs) are graphs containing undirected, directed and bidirected edges but with some additional restrictions on what structures these can take. Note that although these PGM classes are called CGs they are not CGs in the traditional sense since they contain three types of edges, hence when we refer to CGs we only include LWF, AMP and MVR CGs.

If we go back to our example in Figure 2.1 we can see that the graph in Figure 2.1a is not an LWF CG, AMP CG, MVR CG or MAMP CG since it contains a semi-directed cycle, nor an AG since it contains the subgraph $v_2 \leftrightarrow v_4 - v_3$. An example of an LWF CG or an AMP CG H is shown in Figure 2.2a, while an example of an MVR CG F is shown in Figure 2.2b. We can here see that H contains three chain components $\{v_1\}$, $\{v_2\}$ and $\{v_3, v_4\}$ and that F contains two chain components $\{v_1\}$ and $\{v_2, v_3, v_4\}$.

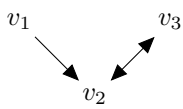
Let X, Y, Z and W denote four disjoint subsets of V. We say that X is *conditionally independent* from Y given Z if the value of X does not influence the value of Y when the values of the variables in Z are known, i.e. $p(X, Y|Z) = p(X|Z)p(Y|Z)$ holds whenever $p(Z) > 0$. We denote this by $X \perp_p Y|Z$ if it holds in a probability distribution p while with $X \not\perp_p Y|Z$ we mean that it does not hold in 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 this by $X \perp_G Y|Z$ and we will discuss different separation criteria for CGs in Chapter 4. Similarly we denote with $X \not\perp_G Y|Z$ that the separation criterion of G does not represent

the conditional independence.

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). 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 $X \perp_M Y|Z \wedge X \perp_M W|Z \Rightarrow X \perp_M Y \cup W|Z$.

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. 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 $I(G) \subset I(H) \subseteq I(p)$.

To illustrate the last concepts we can look at the MVR CG J and the independence models in Figure 2.3. In Figure 2.3b we can see the independences that hold in J and hence the independence model of J. Finally we can also see another independence model in Figure 2.3c such that $I(J) \subseteq M$.



(a) An MVR CG J.

$v_1 \perp_J v_3|\varnothing$
$v_3 \perp_J v_1|\varnothing$

(b) The independence model of J.

$v_1 \perp v_3|\varnothing$
$v_3 \perp v_1|\varnothing$
$v_1 \perp v_3|v_2$
$v_3 \perp v_1|v_2$

(c) Another independence model M.

Figure 2.3: Example of independence models.

Chapter 3

CGs' relation to other PGM classes

PGM classes differ in the types of edges and structures they can contain as well as their separation criteria, hence they differ in what systems and independence models they can represent. Depending on the independence models a PGM class can represent we can discuss its expressivity and we say that a PGM class is more expressive than another class if it can express more independence models.[1] It can be noted that this is not the same as representable graphs, since it is often the case that multiple graphs represent the same independence model. The more basic PGM classes, such as DAGs and UGs, can represent relatively few independence models for any number of nodes and hence are not so expressive. On the other hand, the more general PGM classes, such as AGs, can represent a relatively large number of independence models and hence are more 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 make assumptions, for example that only causal relations exist between the 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 compared to less expressive classes. Graphs with multiple types of edges can also be harder to reason about and interpret since the interpretation of what an edge represents is not always clear. Because of this, the more basic classes, such as DAGs and UGs, have received more attention in research and hence more efficient learning and inference algorithms exist for these compared to the more general classes.

[1] In this thesis we only discuss expressivity as the number of representable independence models, not in terms of causal structures etc.

It is often the case that a PGM class subsumes another PGM class in terms of expressivity, meaning that every independence model represented by the subsumed PGM class also can be represented as the subsuming PGM class. For example, a CG containing only directed edges is actually a DAG, which means that any independence model that can be represented by a DAG can be represented by a CG. Similarly, any independence model represented by an UG (resp. BG) can be represented by an LWF or AMP CG (resp. MVR CG). In Figure 3.1[2] the subsumption relations are shown between different well-studied PGM classes. We can see here that all CGs are loopless graphs, but apart from this they do not share any other superclass. MVR CGs are, however, a subclass of regression CGs, introduced by Wermuth and Sadeghi [38], 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 [21]. 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 are 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 DAGs, such as for example plate models, temporal models and efficient inference, learning and parametrization algorithms [11], can be extended to other PGM classes, and this extension is more straightforward for PGM classes that are similar to DAGs such as CGs. This is true both for finding methods for extending the ideas and algorithms from DAGs and for proving their correctness. Moreover, as we have used existing results from UGs and DAGs to extend ideas to CGs, we hope that our results will allow others to extend these ideas further to more expressive PGM classes, such as MAMP CGs and AGs.

[2]For PGM classes not defined in this thesis please see the work by Sadeghi [26] and Peña [21].

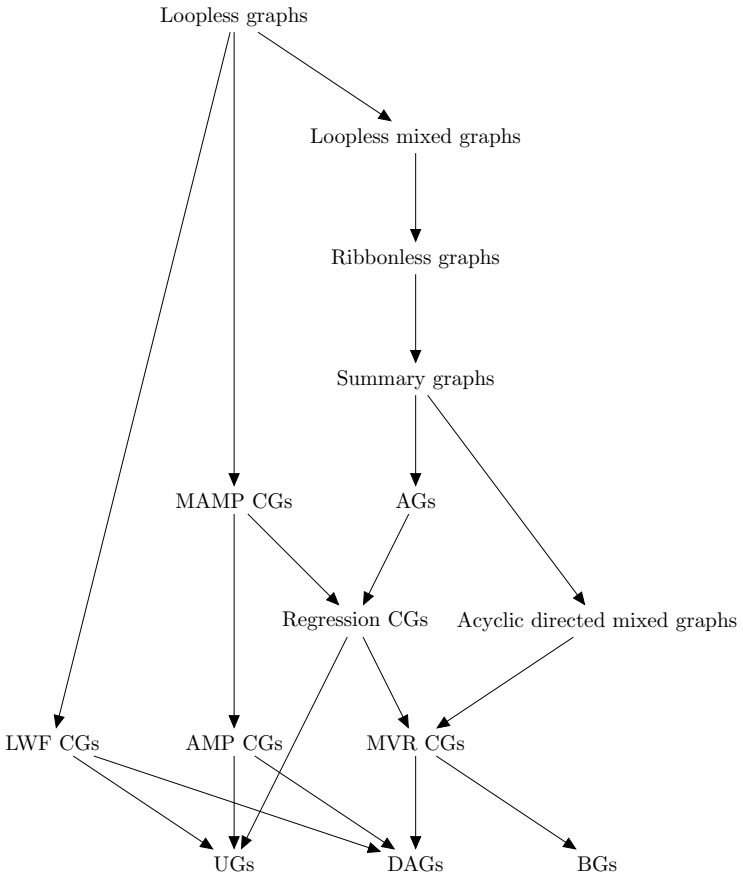


Figure 3.1: The hierarchy of PGM classes.

Chapter 4

The CG interpretations

As noted in the introduction, the different CG interpretations share the characteristic of having chain components that are connected to each other by directed edges while having a second, symmetric, type of edge internally. However, it does not directly follow what kind of symmetric relation this second type of edge represents, nor how the combination between it and the directed edges should be interpreted. Hence, several different interpretations have been proposed, each with its own advantages and disadvantages, and where no interpretation clearly outperforms another.

In this chapter we introduce and discuss these CG interpretations in more detail and try to clarify the following concepts and questions for each of them:

- Separation criterion: What is the separation criterion for the graphs?
- Parametrization: What parametrizations exist for discrete respectively continuous variables?
- Intuitive meaning: What is the intuitive meaning of the edges in the graphs?

In addition to this we also give examples of how the interpretations can be used. First, however, we will describe what these concepts are in more detail, why they are interesting, and how they apply to DAGs.

Separation criterion

The separation criterion defines which independences are represented by a graph and thereby its independence model. For any given PGM class it often exists multiple ways of describing this criterion, but in this thesis we focus on the criterion that operates directly on the graph, as opposed to other existing criteria that first transform the graph and then calculate the represented independences. For DAGs the separation criterion is as follows.

Given three disjoint sets of nodes X, Y and Z in a DAG 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 v_j is said to be a *collider* between two nodes v_i and v_k on a path if the following configuration exists in the path: $v_i \to v_j \leftarrow v_k$. For any other configuration the node v_j is a non-collider on the path.

Parametrization

As noted in the introduction, a PGM induces a factorization of a (joint) probability distribution into several smaller probability distributions, which is beneficial because, for example, the independences are illuminated and calculations can be done faster. For different PGM classes these factorizations are performed differently but for a DAG G, with variables V, a joint probability distribution $p(V)$ is factorized into smaller probability distributions where each node $v_i \in V$ is only dependent on the parents of that node, i.e. $p(V) = \prod_{\forall v_i \in V} p(v_i|pa_G(v_i))$.

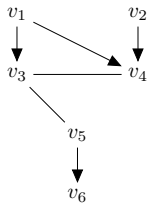
For example, according to the DAG shown in Figure 1.1, the probability distribution $p(Rain, WetStreet, WetLawn)$ can be factorized as $p(WetStreet|Rain)p(WetLawn|Rain)p(Rain)$. These factorizations can then be used to efficiently parametrize the graphs. In Figure 1.1b we saw how this was done for discrete variables using probability tables, while for continuous variables, with normally distributed errors, it is typically represented as a system of linear equations. In the associated system of linear equations to a DAG G each node $v_i \in V$ is modelled by the equation $v_i = \beta_i \cdot pa_G(v_i) + \epsilon_i$, where β_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.

Intuitive meaning

As noted in the introduction, different PGM classes give different intuitive meaning to their edges. For DAGs the intuitive meaning is simply that the parent nodes are the cause of the child nodes, but we will see that the intuitive meaning for CGs can be harder to describe.

4.1 The LWF interpretation

The LWF CG interpretation was introduced by Lauritzen, Wermuth and Frydenberg in 1989 [9, 14] and is the most well-researched CG interpretation. As noted above, LWF CGs contain undirected 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:



(a) An example CG G.

$v_1 \perp_G v_2 | \varnothing$
$v_2 \not\perp_G v_3 | \varnothing$
$v_1 \not\perp_G v_2 | v_3$
$v_1 \perp_G v_6 | v_5$
$v_2 \not\perp_G v_5 | v_4$

(b) LWF CG separations in G.

$v_1 \perp_G v_2 | \varnothing$
$v_2 \perp_G v_3 | \varnothing$
$v_1 \perp_G v_2 | v_3$
$v_1 \perp_G v_6 | v_5$
$v_2 \not\perp_G v_5 | v_4$

(c) AMP CG separations in G.

Figure 4.1: An example CG G and some corresponding separations according to the LWF and AMP interpretations.

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 $v_i, \dots, v_{i+n}$ such that the route contains the subroute $v_i - v_{i+1} - \dots - v_{i+n}$. It is called a *collider section* if $v_i \dots v_{i+n}$ together with the two neighboring nodes in the route, v_{i-1} and v_{i+n+1} (note that v_{i-1} and v_{i+n+1} might be the same node), form the subroute $v_{i-1} \to v_i - v_{i+1} - \dots - v_{i+n} \leftarrow v_{i+n+1}$ in the route. For any other configuration the section is a non-collider section.

A simple example of a CG is shown in Figure 4.1a. Here the CG has four chain components: $\{v_1\}$, $\{v_2\}$, $\{v_3, v_4, v_5\}$ and $\{v_6\}$. If the graph is interpreted as an LWF CG the separations and non-separations shown in Figure 4.1b hold. Note that these are not all the separations that hold in G.

For CGs the different interpretations have different separation criteria, which means that they represent different factorizations and require different parametrizations. However, the chain components can, to some degree, be seen as supernodes in the graphs and hence can be treated similarly to nodes in a DAG. This means that the factorization of a joint probability distribution $p(V)$ according to a CG G, with chain components $K_1, \dots, K_m$, is $p(V) = \prod_{i=1,\dots,m} p(K_i | pa_G(K_i))$. As an example of this the CG shown in Figure 1.2b represents the factorization
$p(Sprinkler, StreetCleaned, WetLawn, WetStreet)$
$= p(WetLawn, WetStreet | Sprinkler, StreetCleaned) p(Sprinkler) p(StreetCleaned)$.

To parametrize a CG G this factorization can be used. In the discrete case the conditional probability distributions $p(K_i | pa_G(K_i))$ can be parametrized directly as a joint probability distribution over the nodes in K_i. Moreover, for some of the interpretations these conditional probability distributions can also be broken down further into smaller distributions. In the case of LWF CGs each component K_i can, for example, be factorized clique-wise as $p(K_i | pa_G(K_i)) = \frac{1}{Z_i} \prod_{M \in M_C} \phi_M$, where M_C are the cliques in the *closure graph* of K_i. The closure graph of K_i is the induced subgraph

$G_{K_i \cup pa_G(K_i)}$ where each directed edge is replaced by an undirected edge and each pair of vertices in $pa_G(K_i)$ is also connected by an undirected edge. Each ϕ_M is then a 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 an UG and be parametrized as such when discrete variables are used. A more detailed description of how this is performed is written by Peña [19], where he also describes what form the parameters must take to be unique for a given joint probability distribution.

In the case of continuous variables, 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 DAG. Hence the linear equation of a node v_j in a CG is $v_j = \beta_j \cdot pa_G(K_i) + \epsilon_j$ where K_i is the component to which v_j belongs [31]. Here the β_j-vector represents the influence of the parents of the component over the nodes in the component while ϵ_j represents the noise, or influence, between the nodes in the same component. What differs between the parametrizations of the different CG interpretations is the interpretation and modelling of β_j and ϵ_j given G.

For example, in the case of LWF CGs the k-th element of β_j can be interpreted as the sum of the weights of all the paths in G between the parent x_k of K_i and the node v_j in K_i such that the nodes in these paths are all in $x_k \cup K_i$. The weight of each path is also the product of the weights of its edges [31]. Meanwhile, the noise ϵ_j in an LWF CG is determined by the associated inverse covariance matrix of that component such that an entry in the inverse covariance matrix for two nodes v_j and v_m can be non-zero iff there exists an undirected edge $v_j - v_m$ in G. For example, from Figure 4.1a it follows that the influence from node v_2 onto node v_4 is direct since only one path exists between them. However, the influence from node v_1 onto node v_5 is determined by the path $v_1 \to v_3 - v_5$ as well as $v_1 \to v_4 - v_3 - v_5$. This characterization 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 v_1 onto v_5 is the same as v_1 onto v_3 except for the last path between v_3 and v_5.

An example of a situation when LWF CGs are useful is when we want to model a system with knowledge obtained from several experts, each with his or her own exclusive field of competence. Each expert then gives information about the structural relationships between the variables within his or her domain given outside factors that affect the variables in his or her domain of expertise. The expert does this by providing an UG over the variables in the domain and their outside factors. Moreover, since the expert only knows about his or her domain and not how the outside factors are related, he or she must assume that all outside factors are adjacent to each other when creating the UG. The subgraph of the UG induced by the variables in the experts' domain can then be seen as a component in a resulting LWF CG

while the outside factors are added as parents to their previous neighbors in the component. The internal structure of the outside factors will be defined by some other expert, who is an expert in that domain. If a strict causal ordering is maintained between the variables, putting the different chain components together into a single graph then results in an LWF CG [37]. An example of this would be if we have three medical experts, one expert modelling the probability that a person will have certain gene-expressions in his or her DNA, one that models the probability of different protein signalling data occurring in blood samples given these gene-expressions, and one that models the occurrence of different traits, such as diseases, given the gene-expressions.

Other settings in which LWF CGs are appropriate to use is when the variables of a system can only be measured in an aggregated state [8] or when modelling the equilibrium state of a system containing feedback loops [13]. An example of this has been given by Lappenschaar et al. [12] where they used LWF CGs to model the interaction between diabetes and lipid disorder given the relevant factors. In this approach the authors propose a qualitative parametrization where it was only calculated whether two adjacent variables in the graph had positive, negative or ambiguous influence on each other, and not the actual parameter value. Finally, it is also worth noting that if an LWF CG only contains directed edges it can be read as a DAG while if it only contains undirected edges it can be read as an UG.

4.2 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 some component-wise recursive characteristics of DAGs. Similarly to LWF CGs, AMP CGs also contain components connected to each other by directed edges, whereas each component internally contains only undirected edges. As a result, an AMP CG containing only directed edges can be read as a DAG and an AMP CG containing only undirected edges can be read as an UG, similarly to an LWF CG. 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 no route exists between X and Y such that:

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

A node v_j is said to be a *collider* in an AMP CG G between two nodes v_i and v_k on a route if one of the following configurations exists in G: $v_i \rightarrow v_j \leftarrow v_k$, $v_i \rightarrow v_j - v_k$ or $v_i - v_j \leftarrow v_k$. For any other configuration the node v_j is a non-collider. In the case of the CG shown in Figure 4.1a, we can see that the separations and non-separations in Figure 4.1c hold if we interpret it as an

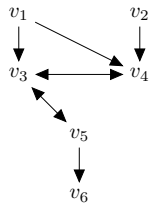
AMP CG. Note that these are not all the separations and non-separations that hold in G.

Unlike the case of LWF CGs no factorization of the internal structure of a chain component of an AMP CG exists today. This is because AMP CGs may induce non-smooth models as shown by Drton [7]. Hence, an AMP CG G, with the variables V and chain components $K_1, \ldots, K_m$, only represents the factorization $p(V) = \prod_{i=1,\ldots,m} p(K_i|pa_G(K_i))$. This means that for any discrete parametrization each probability distribution $p(K_i|pa_G(K_i))$, $1 \leq i \leq m$, has to be kept intact and cannot be factorized further into smaller distributions.

In the case of continuous variables, the associated system of linear equations of an AMP CG also differs from an LWF CG in the way the noise is modelled. For a node v_j, in an AMP CG G, the associated linear equation is $v_j = \beta_j \cdot pa_G(v_j) + \epsilon_j$ and hence the node depends only on its parents and not on the parents of the whole component, as it does in the case of LWF CGs [31]. The noise ϵ_j is then controlled by the inverse covariance matrix of that component where the corresponding entry in the inverse covariance matrix for two nodes v_k and v_l can be non-zero iff there is an undirected edge $v_k - v_l$ in G. 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 v_3 and v_4 in Figure 4.1a block the influence from the parents v_1 and v_2 onto v_5 if the graph is interpreted as an AMP CG.

AMP CGs are useful when we have a set of variables with no causal ordering, so the relations should be modelled as a UG, but also a second set of variables which can be seen as causes for some of these variables in the first set. The internal structure of the first set of variables can then be modelled as an UG, creating a chain component in an AMP CG, and the causes as parents of some of the variables in the chain component. Note that for AMP CGs the parents only affect the direct children in the chain component, not all the nodes in the component such as in the case of LWF CGs. An example in medicine where such a model might be appropriate is the modelling of pain levels of different areas on the body of a patient. The pain levels can then be seen as correlated "geographically" over the body, and hence can be modelled as an UG. Certain other factors do exist, however, which alter the pain levels locally at some of these areas, such as the type of body part the area is located on, if local anaesthetic has been administered in that area, and so on. These outside factors can then be modelled as parents affecting the pain levels locally.

While both LWF CGs and AMP CGs consist of UGs as chain components they differ in the way the parents of the component affect the variables in the component. In an LWF CG each parent affects all the variables in the component, i.e. the information travels through the children, while in an AMP CG the parents only affects the actual children, i.e. the information does not travel to the other variables in the chain component.



(a) Example CG G.

$v_1 \perp_G v_2 | \varnothing$
$v_2 \perp_G v_3 | \varnothing$
$v_1 \perp_G v_2 | v_3$
$v_1 \perp_G v_6 | v_5$
$v_1 \not\perp_G v_6 | v_3$

(b) MVR CG separations in G.

Figure 4.2: An MVR CG and some corresponding separations.

4.3 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 [24]. Cox and Wermuth represented these graphs using directed edges and dashed edges, but we follow Richardson [24] as we feel that the notation is closer to that of DAGs when it comes to the separation criterion.

The most important difference about MVR CGs when compared to AMP CGs and LWF CGs is that MVR CG components contain bidirected instead of undirected edges. As a result, MVR CGs is a superclass of DAGs and BGs instead of DAGs and UGs as in the case of AMP and LWF CGs [4]. MVR CGs also have the following separation criterion: Given three disjoint subsets of nodes X, Y and Z in an MVR CG G, $X \perp_G Y | Z$ iff no path exists 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 $san_G(Z)$.

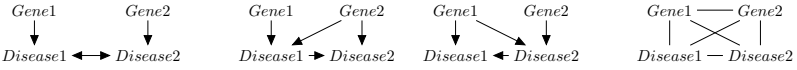
A node v_j is said to be a *collider* in an MVR CG G between two nodes v_i and v_k on a path iff one of the following configurations exists in the path: $v_i \rightarrow v_j \leftarrow v_k$, $v_i \rightarrow v_j \leftrightarrow v_k$, $v_i \leftrightarrow v_j \leftarrow v_k$ or $v_i \leftrightarrow v_j \leftrightarrow v_k$. For any other configuration the node v_j is said to be a non-collider. An example of an MVR CG is shown in Figure 4.2a, with some of the corresponding separations and non-separations in Figure 4.2b.

The discrete parametrization of an MVR CG G, with components $K_1, \ldots, K_m$, is similar to that of an LWF CG in the sense that each conditional probability distribution for $p(K_i | pa_G(K_i))$ can be broken down into smaller parameters. However, unlike LWF CGs, MVR CGs are parametrized using *Möbius parameters*, which makes things slightly more complicated. First, it requires, for each component K_i, that the conditional probability distributions for each subset σ, $\varnothing \subset \sigma \subseteq K_i$ given $pa_G(K_i)$ are specified. In principle, this should require more parameters than specifying

$p(K_i|pa_G(K_i))$ directly, but due to the independence structure of the components it turns out that the larger probability distributions are often factorizations of smaller distributions and hence can be omitted. How this is done in detail is described by Drton [7] and will not be covered in depth in this thesis.

For continuous variables the parametrization is more straightforward. The associated system of linear equations for an MVR CG G is similar to that of AMP CGs: each node depends only on its parents and not on the parents of the whole component. Therefore, the associated linear equation for a node v_j can be written as $v_j = \beta_j \cdot pa(v_j) + \epsilon_j$, where ϵ_j is dependent on the other nodes in the same component. Unlike AMP CGs, MVR CGs can contain non-zero values in the corresponding covariance matrix (not the inverse covariance matrix as for AMP CGs) only for nodes that are spouses [31]. The intuitive meaning behind MVR CGs is therefore very close to that of AMP CGs, differing only in the noise modelling.

A typical situation that gives rise to an MVR CG is when there are hidden variables, i.e. unobserved variables that are parents of at least two observed variables in the data. An example of a situation for which an MVR CG would be useful is if we have a system containing two genes and two diseases caused by these such that *Gene1* is the cause of *Disease1* and *Gene2* is the cause of *Disease2*, but where we also can see that the diseases are correlated. In this case we might 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 in our model, but due to difficulties this would imply [27], let us assume we do not. The MVR CG representing the information above is shown in Figure 4.3a while the inclusion optimal DAGs and UG are shown in Figure 4.3b and 4.3c, respectively. We can now see that it is only the MVR CG that describes the relations in the system correctly.



(a) Inclusion optimal CG. (b) Inclusion optimal DAGs. (c) Inclusion optimal UG.

Figure 4.3: A gene and disease example with MVR CG representation, DAG representations and UG representation.

Chapter 5

Expressiveness

In this chapter we study how many CG models, i.e. independence models representable by CGs, there exist for the different CG interpretations for different numbers of nodes. Doing this allows us to see just how expressive CGs are compared to their subclasses, i.e. how large the benefit is of using this more expressive class of models. It also allows us to calculate other interesting ratios such as the number of CGs per CG model, i.e. how many CGs that represent the same independence model, as well as how often the different CG interpretations intersect in terms of representable independence models.

Counting the number of representable independence models is no easy task since no iterative or closed form expression for this have been presented. Nor is it possible to enumerate every possible CG model for more than five or six nodes due to the exponential increase in models as the number of nodes increases. Nevertheless, one approximate approach that has been shown to give accurate results is to calculate the ratio of independence models representable by CGs that can also be represented as DAGs and then, since every DAG is a CG and the approximate number of DAG models is known, calculate the number of CG models [18]. Furthermore, this ratio can be approximated using a subset of models and, if this subset is sampled uniformly from the whole set of CG models, can be used to represent all CG models of the interpretation under study.

Sampling CG models uniformly is possible using a Markov chain Monte Carlo (MCMC) approach. The approach consists of creating a Markov chain, whose states are the different CG models, and transition between these states using a set of operators. If the operators then fulfill the aperiodicity, irreducibility and reversibility criteria, and k transitions are performed before sampling a state, each state has equal probability of being sampled when $k \to \infty$. This approach has been successfully applied to all three CG interpretations [18, 32, 33] and the results presented in this chapter are based on it. In Section 5.1 we first discuss how good the approximations are and

how expressive the CG interpretations are compared to their subclasses. In Section 5.2 we then discuss the number of CGs per CG model, and finally in Section 5.3 we study how the different CG interpretations intersect in terms of representable independence models.

5.1 Ratios of CG models representable as subclasses

The first thing we study is how expressive CGs are compared to their subclasses. If CGs can only represent a few more models than their subclasses, then it might be argued that the cost of complexity is not worth the gain in expressivity, while the opposite can be argued if they are much more expressive. We start by comparing the number of DAG models with CG models, and these results are shown in Table 5.1. The exact ratios are found by enumerating every possible CG model for the given number of nodes and interpretation and then calculating the ratio of these models that can be represented as DAGs, while the approximate ratios are found as described above. For each number of nodes and interpretation a subset of 10^5 models were sampled with 10^5 transitions between each sampled model. In the table we can note that the approximations are accurate for up to five nodes while for more than five nodes exact enumeration is infeasible. However, by plotting the ratios of the number of DAG models to CG models in a graph with logarithmic scale, as seen in Figure 5.1, we can see that the ratios are linear in the logarithmic scale and hence exponential in a linear scale. More specifically, the equations are $R_{LWF} = 9.1 \cdot 0.654^n$, $R_{AMP} = 7.2 \cdot 0.645^n$ and $R_{MVR} = 6.2 \cdot 0.653^n$, where n is the number of nodes and R the ratio of the subscripted interpretation. This means that the ratios decrease exponentially as the number of nodes increases for all three CG interpretations and that DAGs can only express a tiny fraction of the CG models with only 20 nodes.

We can also compare the number of CG models of the different interpretations to the number of models of their non-directed subclasses, i.e. UGs for LWF and AMP CGs and BGs for MVR CGs. The exact and approximate ratios of these comparisons are shown in Table 5.2. Here we can note that the number of independence models representable by these subclasses is almost non-existent in comparison to the number of CG models for models with 10 or more nodes. Moreover, since the ratios decrease so quickly, we have not been able to find any equations describing the ratios given the number of nodes.

Table 5.1: Exact and approximate ratios of CG models whose independence models can be represented as DAGs.

NODES	EXACT			APPROXIMATE		
	LWF	AMP	MVR	LWF	AMP	MVR
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	0.9250	0.8393	0.8259	0.9327	0.8392	0.8235
5	0.7624	0.6113	0.5905	0.7646	0.6136	0.5900
6				0.5829	0.4382	0.4099
7				0.4179	0.3058	0.2868
8				0.2860	0.2067	0.1951
9				0.1924	0.1407	0.1307
10				0.1286	0.0948	0.0866
11				0.0831	0.0616	0.0565
12				0.0554	0.0403	0.0377
13				0.0349	0.0257	0.0239
14				0.0237	0.0155	0.0159
15				0.0152	0.0108	0.0098
16				0.0096	0.0066	0.0064
17				0.0062	0.0045	0.0049
18				0.0038	0.0028	0.0027
19				0.0027	0.0018	0.0019
20				0.0017	0.0010	0.0011

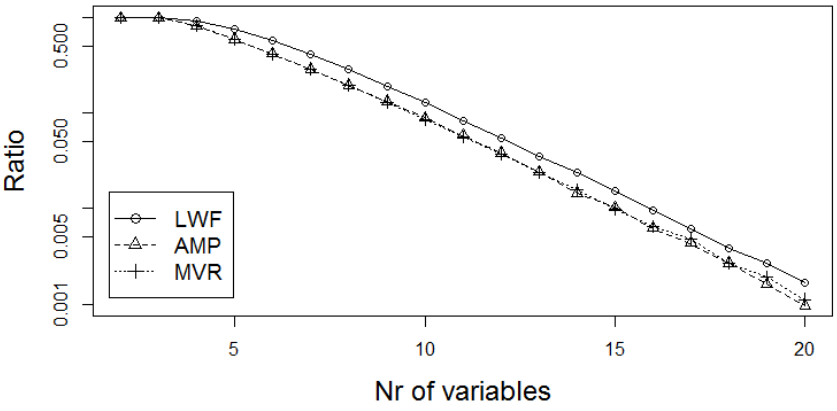


Figure 5.1: The ratios (displayed with a logarithmic scale) of the number of DAG models compared to the number of CG models for different numbers of nodes for the different CG interpretations.

Table 5.2: Exact and approximate ratios of LWF, AMP, and MVR CG models whose independence models can be represented as UGs respectively BGs.

NODES	EXACT			APPROXIMATE		
	LWF	AMP	MVR	LWF	AMP	MVR
2	1	1	1	1	1	1
3	0.7273	0.7273	0.7273	0.7188	0.7275	0.7255
4	0.3200	0.2857	0.2857	0.3122	0.2839	0.2855
5	0.0889	0.0689	0.0689	0.0809	0.0632	0.0697
6				0.0165	0.0112	0.0124
7				0.0032	0.0019	0.0019
8				0.0003	0.0002	0.0003
9				0.0002	0.0000	0.0000
10				0.0000	0.0000	0.0000

5.2 Ratios of the number of CGs per CG model and approximate number of representable models

For CGs, as for many other PGM classes, multiple graphs can represent the same independence model, even if they are interpreted using the same CG interpretation. Therefore, an interesting question is how many CGs there are per CG model on average. These ratios are shown in Table 5.3 and can be found using the equation

$$\frac{\#CGs}{\#CGmodels} = \frac{\#CGs}{\#DAGs} \cdot \frac{\#DAGs}{\#DAGmodels} \cdot \frac{\#DAGmodels}{\#CGmodels} \tag{5.1}$$

where $\#CGmodels$ represents the number of CG models of a certain interpretation and so on. The ratio $\frac{\#CGs}{\#DAGs}$ can then be found using the iterative equations by Robinsson [25] and Steinsky [35] while $\frac{\#DAGs}{\#DAGmodels}$ has been approximated in previous studies for DAG models [18]. Finally, we can also get the ratio $\frac{\#DAGmodels}{\#CGmodels}$ from Table 5.1. If we study the values in Table 5.3 we can see that the average number of CGs per CG model appears to converge to approximately 26 for the LWF CG interpretation and 17 for the AMP and MVR CG interpretations. This corresponds well with what we have seen for DAGs, although in that case the convergence was around 4 DAGs per DAG model [18]. This means that traversing the space of CG models when learning CG structures is considerably more efficient than traversing the space of all CGs. However, at the same time, it also means that this efficiency does not scale as the number of nodes in the graphs increases.

Finally, in Table 5.4, we can see the number of CG models for the different CG interpretations. These numbers follow directly from the number of

Table 5.3: Exact and approximate numbers of CGs per CG model.

NODES	EXACT			APPROXIMATE		
	LWF	AMP	MVR	LWF	AMP	MVR
2	2	2	2	1.97	1.97	1.97
3	4.55	4.55	4.55	4.47	4.47	4.47
4	8.44	7.54	7.54	8.61	7.75	7.60
5	12.38	9.59	9.59	12.61	10.12	9.73
6				15.80	11.87	11.11
7				18.05	13.21	12.39
8				20.20	14.59	13.77
9				20.97	15.34	14.25
10				22.61	16.66	15.23
11				23.14	17.16	15.74
12				23.66	17.22	16.09
13				22.88	16.85	15.64
14				24.64	16.10	16.54
15				25.63	18.20	16.60
16				24.87	17.15	16.63
17				24.94	18.37	19.67
18				24.24	17.89	16.94
19				26.51	17.38	18.96
20				26.26	16.29	17.72

CGs per CG model, shown in Table 5.3, and the equations for calculating the number of CG structures for a given number of nodes defined by Steinsky [35]. We can here see that the AMP and MVR CG interpretations can represent approximately the same number of independence models, while the LWF CGs only can represent approximately 65% of this number since LWF CG Markov equivalence classes are larger on average. Hence the AMP and MVR interpretations are the most expressive CG interpretations, in terms of the number of representable independence models, while the LWF interpretation falls behind. The ratio between them does, however, appear to be constant as the number of nodes in the models increases. This also follows from the observation above that the average Markov equivalence class sizes were constant for the different CG interpretations.

5.3 Intersections between the CG interpretations in terms of representable independence models

In Table 5.1 we saw the ratio of CG models whose independence model could be represented by DAGs. The representable independence models for the different CG interpretations do, however, intersect over more models than this subclass can represent. For example, UGs exist whose independence models are representable as both AMP CGs and LWF CGs, but not

Table 5.4: Exact and approximate numbers of CG models representable for the different CG interpretations.

NODES	EXACT			APPROXIMATE		
	LWF	AMP	MVR	LWF	AMP	MVR
2	2	2	2	2.03	2.03	2.03
3	11	11	11	11	11	11
4	200	224	224	196	218	222
5	11519	14869	14866	11313	14097	14662
6				1.83 E+6	2.43 E+6	2.60 E+6
7				7.57 E+8	1.03 E+9	1.10 E+9
8				7.31 E+11	1.01 E+12	1.07 E+12
9				1.71 E+15	2.34 E+15	2.52 E+15
10				8.57 E+18	1.16 E+19	1.27 E+19
11				9.95 E+22	1.34 E+23	1.46 E+23
12				2.53 E+27	3.47 E+27	3.71 E+27
13				1.47 E+32	1.99 E+32	2.15 E+32
14				1.65 E+37	2.52 E+37	2.46 E+37
15				4.11 E+42	5.79 E+42	6.35 E+42
16				2.34 E+48	3.40 E+48	3.50 E+48
17				2.75 E+54	3.73 E+54	3.48 E+54
18				7.04 E+60	9.53 E+60	1.01 E+61
19				3.38 E+67	5.16 E+67	4.73 E+67
20				3.78 E+74	6.09 E+74	5.60 E+74

as DAGs. The graphical conditions for when the independence model of a CG of one interpretation can be represented by another interpretation have been defined and proven correct [1, 30]. Hence, using the sampled CG models we can estimate the sizes of the different intersections. The results of this are shown in Table 5.5 where the number of independence models in an intersection is compared to the number of all representable models for the relevant CG interpretation. The ratios are also illustrated in Figure 5.2 to allow the reader a better overview of how the spaces of the intersections and representable independence models change as the number of nodes in the models increases. We can here see that almost all independence models representable by LWF CGs can only be represented by this interpretation while the intersection between the AMP and MVR CGs is quite large (25% for 20 nodes). This means that, although AMP and MVR CGs share some models, almost all models representable by each CG interpretation are unique to that interpretation, and thereby that the different CG interpretations model different types of systems.

Table 5.5: Exact and approximate ratios of CG models that intersect between the different CG interpretations.

NODES	**Ratio of LWF representable as**		**Ratio of AMP representable as**		**Ratio of MVR representable as**	
	AMP	MVR	LWF	MVR	LWF	AMP
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	0.9327	0.9327	0.8392	0.9300	0.8235	0.9126
5	0.7744	0.7646	0.6215	0.8303	0.5900	0.7984
6	0.5997	0.5829	0.4507	0.7519	0.4099	0.7033
7	0.4344	0.4179	0.3178	0.6845	0.2868	0.6419
8	0.2987	0.2860	0.2159	0.6346	0.1951	0.5990
9	0.2011	0.1924	0.1470	0.5914	0.1307	0.5495
10	0.1344	0.1286	0.0990	0.5432	0.0866	0.4966
11	0.0866	0.0831	0.0642	0.5027	0.0565	0.4613
12	0.0580	0.0554	0.0422	0.4688	0.0377	0.4382
13	0.0365	0.0349	0.0269	0.4352	0.0239	0.4040
14	0.0248	0.0237	0.0162	0.3982	0.0159	0.4092
15	0.0158	0.0152	0.0112	0.3718	0.0098	0.3390
16	0.0101	0.0096	0.0069	0.3453	0.0064	0.3349
17	0.0063	0.0062	0.0047	0.3164	0.0049	0.3387
18	0.0040	0.0038	0.0030	0.2928	0.0027	0.2772
19	0.0028	0.0027	0.0019	0.2759	0.0019	0.3011
20	0.0018	0.0017	0.0011	0.2507	0.0011	0.2726

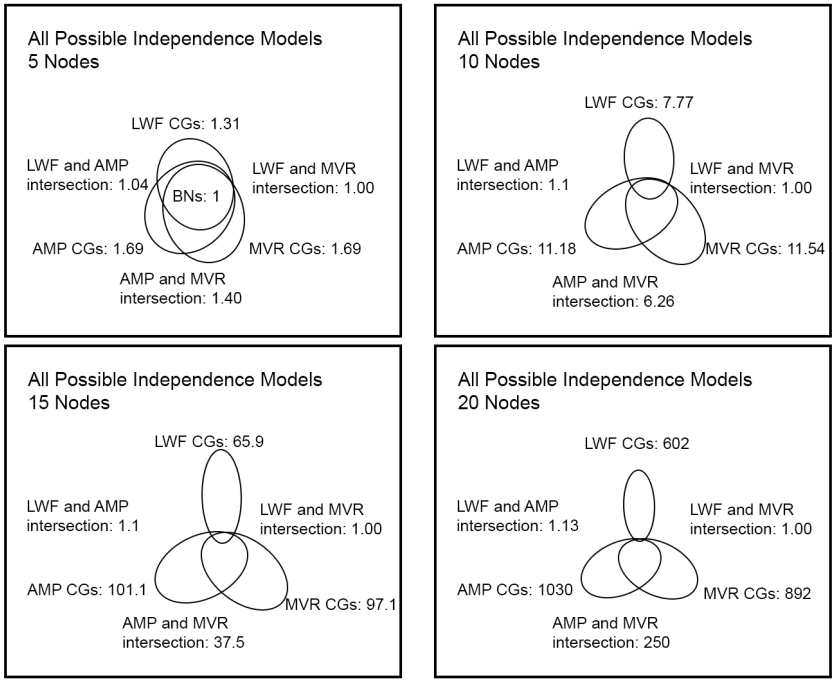


Figure 5.2: The intersections of representable independence models for the different CG interpretations for different number of nodes in the models. In the figures the space of independence models representable by DAGs have size 1 and all other spaces are relative to this. For example, LWF CGs can represent 31% more independence models than DAGs for 5 nodes while all independence models representable by both LWF and AMP CGs, including the DAGs, are only 4% more than those representable by DAGs.

Chapter 6

Structure learning algorithms

One important aspect of PGMs is the possibility to learn the models directly from data. This data can be in many forms but it is typically represented as a set of samples such that each sample contains the states or values of all variables in the system that is being modelled. In this thesis we will assume all samples are independent and identically distributed and hence that, as the number of samples goes to infinity, the probability distribution induced by the samples is equal to the true probability distribution of the variables in the underlying system. Using such a set of samples, and the induced probability distribution, learning algorithms can then be used to find the graph that best represents the data. Once the graph is found the parameters of the PGM can also be calculated such that the model best represents the data.

In this chapter we will look into the CG structure learning algorithms that, to the authors knowledge, have been presented for CGs. The algorithms differ in their definition of what makes a CG best represent the data, i.e. their optimization criteria, but most algorithms can be categorized into two approaches: The score-based approach that finds the graph that minimizes some score function, typically the penalized log likelihood, of the data given the graph and the constraint-based approach that finds an inclusion optimal graph with respect to the independence model represented by the data. The former approach usually works by making small local changes to the graph of the PGM, iteratively increasing its score, until a maximum has been found. This requires the score, and thereby the parametrization, to be recalculated for each new model. This in turn requires that all calculations can be performed with low complexity, since a large number of models needs to be evaluated. For DAGs this is possible since any change in the graph only requires local parameters to be updated and the parameters can be calculated using closed form equations. Moreover, most score functions

decompose into a product of factors, meaning that only local factors have to be recalculated when evaluating the DAG after each change and hence only a small part of the score, and parametrization, needs to be recalculated for each iteration. However, for CGs this is not possible, and therefore no score-based algorithms have been presented so far.

For the latter approach, the constraint-based approach, four learning algorithms exist for CGs today. This approach is based on testing if variables are (conditionally) independent in the data using an independence test, such as the χ^2 test, and using this information to deduce the structure of the optimal graph. The algorithms we cover in this thesis are the PC-like algorithms [20, 29, 36], the ASP algorithms [22, 28], the LCD algorithm [15] and the CKES algorithm [23]. The former two have implementations for all three CG interpretations, while the latter two are only applicable for LWF CGs.

6.1 The PC-like algorithms

The PC algorithm is a constraint-based structural learning algorithm presented for DAGs [16, 34] and is based on three sequential phases. The first phase consists of finding the adjacencies of the graph. The resulting graph contains an edge $v_1 - v_2$ iff no set $S \subseteq V \setminus \{v_1, v_2\}$ exists such that $v_1 \perp v_2 | S$ holds in the data. In the second phase some of the undirected edges are then oriented into directed edges according to a set of rules. These rules are applied iteratively until no rule is applicable and results in the so-called *essential graph* which contains all directed edges oriented in the same direction in all DAGs in that Markov equivalence class. Finally in the third phase the remaining undirected edges are oriented, transforming the graph into a DAG in the Markov equivalence class.

PC-like algorithms currently exist for all three CG interpretations [20, 29, 36] where the different phases are slightly altered according to the interpretation but the basic ideas are kept the same. The first phase finds the adjacencies, the second orients the edges that must be oriented the same in every CG in the Markov equivalence class and the third phase transforms this graph into a CG. Experimental results have shown that this type of algorithm scales well with the number of nodes [15] and it is proven that these algorithms find the optimal CG if the probability distribution p of the data is faithful to some CG G of the correct interpretation, i.e. if a CG G exists such that $I(G) = I(p)$ [20, 29, 36]. However, if no such CG exists, it cannot be guaranteed that the learnt CG G can factorize the probability distribution, i.e. that $I(G) \subseteq I(p)$ holds [23]. Moreover, in the case of LWF CGs, it may also be the case that the learnt LWF CG contains semi-directed cycles if p is not faithful to some LWF CG.

6.2 The ASP algorithms

The ASP algorithms are based on the idea of encoding the graph learning problem as an answer set programming (ASP) problem and then solving it using state-of-the-art exact ASP optimization solvers. This means that a complete search in the space of possible CGs is performed and currently encodings for all three CG interpretations exist [22, 28].

The CG learning problem is encoded using first order logical rules and the encoding can be divided into three separate parts, (1) the possible CG structure, (2) the represented dependences and independences for any CG structure as well as in the data and (3) a well-defined objective function. Note that parts 2 and 3 are dependent on the input data. Specifically, part 2 represents the dependences and independences that are determined by the data and it might also contain additional information about these such as the confidence of their correctness. This information can then be used by the objective function in part 3. Essentially, by calling an ASP solver with the whole ASP encoding, consisting of parts 1–3, the solver will perform an intelligent implicit search over the space of CGs (using part 1), and will output a CG that produces the best objective function score (based on part 3) by deriving the separations and non-separations in the CGs (using part 2).

Using first order logic to encode the learning problem has many benefits. First of all it is easy to add expert knowledge, such as that the learnt CG should contain a causal relation from one node to another or that a causal ordering should be maintained among the variables. Secondly, it also allows for a range of objective functions to be used. This might, for example, be to find the CG that represents all dependences in the data while at the same time representing as many independences as possible, i.e. the best inclusion optimal CG given the data, or the CG that represents all the dependences in the data, while at the same time containing the minimum number of edges. Another objective function that has been shown to give good results is to assign weights to the dependences and independences derived from the data, depending on how well the data supports the conclusion, and then minimize the sum of the weights of the dependences and independences not represented in the CG [28]. A third benefit of using the ASP paradigm is that even without making any assumptions about the probability distribution of the data, such as that it should be faithful to some CG, the learnt CG is always optimal with respect to the objective function. This is possible since the whole space of CGs is covered as possible solutions. However, searching the whole set of possible solutions is also the source of the ASP paradigm's main disadvantage which is its high complexity. Even though the current state-of-the-art ASP solvers are very efficient, this method is infeasible to use on a modern desktop computer for more than seven nodes unless additional expert knowledge is added, restricting the search space of possible CGs.

6.3 The LCD algorithm

The LCD (learn chain graph via decomposition) algorithm was introduced by Ma et. al [15] to learn LWF CGs and is based on a divide and conquer approach. Just like the PC algorithm, it starts out by finding the adjacencies in the CG. However, unlike the PC algorithm, it achieves this by decomposing the graph into smaller pieces and then reconstructing the adjacency graph by merging the decomposed graphs together. The second phase of the LCD algorithm then uses rules to orient some of the edges in the CG in an efficient manner with lower complexity than the PC-like algorithm for LWF CGs [15]. Empirical simulations have shown that the algorithm achieves competitive results with the other presented CG learning algorithms when the probability distribution of the data is faithful to some LWF CG [23, 28]. However, just as in the case of the PC-like algorithms, unless the probability distribution p of the data is faithful to some LWF CG the learnt CG cannot be ensured to factorize p properly. It should also be noted that the learnt graph of the LCD algorithm is only the pattern of a LWF CG G, i.e. the graph that contains the complexes of G but which may also contain semi-directed cycles unless p is faithful.

6.4 The CKES algorithm

The CKES algorithm was introduced by Peña et. al [23] and is based on the extension of Meek's conjecture to LWF CGs. It states that given two LWF CGs G and H, such that $I(H) \subseteq I(G)$, G can be transformed into H through a set of undirected and directed edge additions, feasible splits and feasible mergings such that after each change $I(H) \subseteq I(G)$ holds. A feasible split and feasible merging are operations that change the structure of a CG without changing its Markov equivalence class, i.e. represented independence model. This allows the CKES algorithm to start from a CG with no edges and iteratively add and remove edges to the CG model, incrementally improving its fit with respect to the independence model of the data. Then, when no improvements are possible, the resulting model is inclusion optimal [23].

The CKES algorithm has several advantages compared to the other learning algorithms. First, the CKES algorithm can be guaranteed to find an inclusion optimal LWF CG as long as the probability distribution of the data fulfills the graphoid and composition properties. This is a considerable relaxation of the faithfulness property, required by the PC-like and LCD algorithms, and can be shown to hold for any Gaussian probability distribution. Secondly, the CKES algorithm scales better than the presented algorithms based on the ASP paradigm, and simulated empirical results have been presented for up to 20 nodes. However, the empirical results also show that the algorithm is overly generous in including edges in the found LWF CGs and that it may not find the best inclusion optimum when multiple inclusion optimal CGs exist [23]. Moreover, it has been shown that

Meek's conjecture does not hold for the AMP and MVR CG interpretations, and hence it is not possible to directly adopt this algorithm to these interpretations.

Chapter 7

Conclusions and future work

In this thesis we have given an introduction to CGs and the research presented in the area so far. Using a more expressive PGM class, such as CGs, compared to a less expressive PGM class, such as DAGs, offers many benefits, most importantly that we are more likely to be able to model a system correctly. This is because we have a much wider range of models to choose from and, as we showed in Chapter 5, only a tiny fraction of the models representable by CGs can be represented by DAGs. Moreover, with the recent advancements in research on CGs, such as learning algorithms and parametrizations, the core components for practical use have been extended from DAGs to CGs making them a viable choice when modelling systems containing both symmetric and asymmetric relations.

With this thesis it is our hope to provide novice users with an overview of the field of CGs and to allow them to apply these models to their problems and domains. This is important since practical applications of CGs are currently still rare in research, with a few exceptions [2, 4, 6]. Applying the models to real world problems would clarify which types of systems are suited to be modelled by CGs and where their advantages are most evident. This in turn would hopefully also better clarify what relations the different CG interpretations can represent and the intuition behind their edges.

Another important goal of this thesis is to introduce researchers in other PGM subfields to CGs, allowing them to consider how their ideas or algorithms can be extended to CGs. Many concepts and ideas of DAGs have already been extended to the research field, but it is still lacking in a number of areas. The most crucial is probably efficient parametrization methods and inference algorithms, both exact and approximate, to allow large CG structures to be parametrized from data and used in practice. Another important area for future research is efficient scoring functions and score-based learning algorithms, since these have been shown to generally outperform

constraint-based algorithms when applied.

Finally we also hope that this work can inspire new ideas and algorithms in the research of more expressive PGM classes than CGs. Some of the algorithms have already been extended to MAMP CGs and AGs, but many still remain. It would for example be interesting to see the relation between the CG interpretations and their superclasses in terms of representable independence models.

Bibliography

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

[2] S. Carroll and V. Pavlovic. Protein Classification using Probabilistic Chain Graphs and the Gene Ontology Structure. *Bioinformatics*, 22:1871–1878, 2006.

[3] D. R. Cox and N. Wermuth. Linear Dependencies Represented by Chain Graphs. *Statistical Science*, 8:204–218, 1993.

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

[5] A. P. Dawid. Beware of the DAG! *JMLR Workshop and Conference Proceedings In Causality: Objectives and Assessment (NIPS 2008 Workshop)*, 6:59–86, 2008.

[6] A. Denev. *Probabilistic Graphical Models.* Nick Carver, 2015.

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

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

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

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

[11] D. Koller and N. Friedman. *Probabilistic Graphcal Models.* MIT Press, 2009.

[12] M. Lappenschaar, A. Hommersom, and P. J. F. Lucas. Qualatitive Chain Graphs and their Application. *International Journal of Approximate Reasoning*, 55:957–976, 2014.

[13] 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.

[14] 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.

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

[16] C. Meek. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of 11th Conference on Uncertainty in Artificial Intelligence*, pages 403–410, 1995.

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

[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 Discrete Case. *International Journal of Approximate Reasoning*, 50:1306–1313, 2009.

[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. Marginal AMP Chain Graphs. *International Journal of Approximate Reasoning*, 55:1085–1206, 2014.

[22] J. M. Peña. Alternative Markov and Causal Properties for Acyclic Directed Mixed Graphs. arXiv:1511.05835 [stat.ML], 2015. Submitted to UAI 2016.

[23] 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*, pages 778–786, 2014.

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

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

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

[27] I. Shpitser, R. J. Evans, T. S. Richardson, and J. M. Robins. Introduction to Nested Markov Models. *Behaviormetrika*, 41:3–39, 2014.

[28] D. Sonntag, M. Järvisalo, J. M. Peña, and A. Hyttinen. Learning Optimal Chain Graphs with Answer Set Programming. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, pages 822–831, 2015.

[29] 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.

[30] D. Sonntag and J. M. Peña. Chain Graph Interpretations and Their Relations Revisited. *International Journal of Approximate Reasoning*, 58:39–56, 2014.

[31] D. Sonntag and J. M. Peña. Chain Graphs and Gene Networks. *In Foundations of Biomedical Knowledge Representation*, pages 159–178, 2015.

[32] D. Sonntag and J. M. Peña. On Expressiveness of the Chain Graph Interpretations. *International Journal of Approximate Reasoning*, 68:91–107, 2016.

[33] D. Sonntag, J. M. Peña, and M. Gómez-Olmedo. Approximate Counting of Graphical Models Via MCMC Revisited. *International Journal of Intelligent Systems*, 30:384–420, 2015.

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

[35] B. Steinsky. Enumeration of Labelled Chain Graphs and Labelled Essential Directed Acyclic Graphs. *Discrete Mathematics*, 270:266–277, 2003.

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

[37] M. Studený. Bayesian Networks from the Point of View of Chain Graphs. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 496–503, 1998.

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

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

Appendix A

Our contribution

This chapter presents our contribution[1] to the research field of CGs. The chapter starts with a short summary of the work, followed by the publications themselves.

A.1 Summary

We have done research in most of the subareas covered in Chapters 4, 5 and 6 with the exception of the separation criteria and discrete parametrization of CGs. For Chapter 4 the research is mostly restricted to a book chapter written as an introduction to how CGs can be used when reasoning about biomedical knowledge. The book itself is an attempt to connect current research in data representation to research practices in the medical community. The chapter is included as Article 1 in Section A.2 and contains an introduction to the field of CGs written from a gene network modelling perspective. It also contains an appendix where the separation criteria of the different CG interpretations are studied in terms of systems of linear equations. Our contribution for the book chapter consists of compiling the information and writing the chapter, while for the appendix our contribution is restricted to MVR CGs.

When it comes to research on representable independence models our contribution is more substantial. It starts with Article 2, presented in Section A.3, where we used the MCMC approach to approximate the ratio of independence models representable by MVR CGs that are also representable by DAGs. To do this we first had to define a unique graphical representation of the MVR CG Markov equivalence classes, i.e. a set of graphs with a one-to-one relationship to the independence models representable by MVR CGs. We call this representation *essential MVR CGs* and in the article we define its characteristics, i.e. what structure the graph must have to be an

[1] The contribution of the author of this thesis.

essential graph, and a transformation algorithm to transform any MVR CG into its unique representation and vice versa. With such a representation we could then present a set of operators fulfilling the aperiodicity, irreducibility and reversibility criteria and thereby sample MVR CG models and calculate the ratio of models representable as DAGs as described in Chapter 5. Our contribution in the article comprised everything regarding CGs, while the part regarding DAGs was performed by the co-authors.

In Article 3, presented in Section A.4, we then perform a study of how different theoretical concepts of DAGs have been extended to the different CG interpretations. More specifically it studies the concepts: unique representations of independence models, the feasible split and merging operators, the conditions for when an independence model of one CG interpretation can be represented as another CG interpretation, and the extension of Meek's conjecture to CGs. Much of the article is based on previous research but we filled in the gaps where some concept had not been extended to some CG interpretation. These new contributions consist mainly of the feasible split and merging operators for AMP CGs and the conditions for when an independence model of one CG interpretation can be represented by another CG interpretation. Our co-authors also show that Meek's conjecture is not extendible to MVR CGs.

The research on representable independence models is then concluded in Article 4, presented in Section A.5, where we define the MCMC operators for AMP CGs and prove that they fulfill the aperiodicity, irreducibility and reversibility criteria. This allowed us to sample models from all CG interpretations and thereby to get the results presented in Chapter 5.

The remaining articles, presented in Sections A.6, A.7 and A.8, introduce, and prove the correctness of, different CG learning algorithms. In Article 5 we present the PC-like algorithm for MVR CGs discussed in Chapter 6 as well as the split and merging operators for MVR CGs that were used to prove the correctness of the algorithm. The CKES algorithm for LWF CGs is then presented in Article 6, but for this article our contribution mainly consisted 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 theoretical algorithm were infeasible to perform in a reasonable time and hence some approximations had to be made. Finally, in Article 7, we introduce the ASP-based approach to learning CGs. Our implementation was restricted to LWF CGs and our contribution was focused on the encoding of the separation criteria, the implementation, and the experimental evaluation. The expert knowledge of the ASP domain, its solvers and most of the encoding of the CG domain constraints were contributed by the co-authors. The idea was then also extended to AMP and MVR CGs by the co-authors in a subsequent article.

Articles

The articles associated with this thesis have been removed for copyright reasons. For more details about these see:
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-125921

Department of Computer and Information Science
Linköpings universitet

Dissertations

Linköping Studies in Science and Technology
Linköping Studies in Arts and Science
Linköping Studies in Statistics
Linköpings Studies in Information Science

Linköping Studies in Science and Technology

No 14 **Anders Haraldsson:** A Program Manipulation System Based on Partial Evaluation, 1977, ISBN 91-7372-144-1.

No 17 **Bengt Magnhagen:** Probability Based Verification of Time Margins in Digital Designs, 1977, ISBN 91-7372-157-3.

No 18 **Mats Cedwall**: Semantisk analys av process-beskrivningar i naturligt språk, 1977, ISBN 91- 7372-168-9.

No 22 **Jaak Urmi:** A Machine Independent LISP Compiler and its Implications for Ideal Hardware, 1978, ISBN 91-7372-188-3.

No 33 **Tore Risch:** Compilation of Multiple File Queries in a Meta-Database System 1978, ISBN 91- 7372-232-4.

No 51 **Erland Jungert:** Synthesizing Database Structures from a User Oriented Data Model, 1980, ISBN 91-7372-387-8.

No 54 **Sture Hägglund:** Contributions to the Development of Methods and Tools for Interactive Design of Applications Software, 1980, ISBN 91-7372-404-1.

No 55 **Pär Emanuelson:** Performance Enhancement in a Well-Structured Pattern Matcher through Partial Evaluation, 1980, ISBN 91-7372-403-3.

No 58 **Bengt Johnsson, Bertil Andersson:** The Human-Computer Interface in Commercial Systems, 1981, ISBN 91-7372-414-9.

No 69 **H. Jan Komorowski:** A Specification of an Abstract Prolog Machine and its Application to Partial Evaluation, 1981, ISBN 91-7372-479-3.

No 71 **René Reboh:** Knowledge Engineering Techniques and Tools for Expert Systems, 1981, ISBN 91-7372-489-0.

No 77 **Östen Oskarsson:** Mechanisms of Modifiability in large Software Systems, 1982, ISBN 91- 7372-527-7.

No 94 **Hans Lunell:** Code Generator Writing Systems, 1983, ISBN 91-7372-652-4.

No 97 **Andrzej Lingas:** Advances in Minimum Weight Triangulation, 1983, ISBN 91-7372-660-5.

No 109 **Peter Fritzson:** Towards a Distributed Programming Environment based on Incremental Compilation, 1984, ISBN 91-7372-801-2.

No 111 **Erik Tengvald:** The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984, ISBN 91-7372- 805-5.

No 155 **Christos Levcopoulos:** Heuristics for Minimum Decompositions of Polygons, 1987, ISBN 91-7870-133-3.

No 165 **James W. Goodwin:** A Theory and System for Non-Monotonic Reasoning, 1987, ISBN 91-7870-183-X.

No 170 **Zebo Peng:** A Formal Methodology for Automated Synthesis of VLSI Systems, 1987, ISBN 91-7870-225-9.

No 174 **Johan Fagerström:** A Paradigm and System for Design of Distributed Systems, 1988, ISBN 91-7870-301-8.

No 192 **Dimiter Driankov:** Towards a Many Valued Logic of Quantified Belief, 1988, ISBN 91-7870-374-3.

No 213 **Lin Padgham:** Non-Monotonic Inheritance for an Object Oriented Knowledge Base, 1989, ISBN 91-7870-485-5.

No 214 **Tony Larsson:** A Formal Hardware Description and Verification Method, 1989, ISBN 91-7870-517-7.

No 221 **Michael Reinfrank:** Fundamentals and Logical Foundations of Truth Maintenance, 1989, ISBN 91-7870-546-0.

No 239 **Jonas Löwgren:** Knowledge-Based Design Support and Discourse Management in User Interface Management Systems, 1991, ISBN 91-7870-720-X.

No 244 **Henrik Eriksson:** Meta-Tool Support for Knowledge Acquisition, 1991, ISBN 91-7870-746-3.

No 252 **Peter Eklund:** An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies, 1991, ISBN 91-7870-784-6.

No 258 **Patrick Doherty:** NML3 - A Non-Monotonic Formalism with Explicit Defaults, 1991, ISBN 91-7870-816-8.

No 260 **Nahid Shahmehri:** Generalized Algorithmic Debugging, 1991, ISBN 91-7870-828-1.

No 264 **Nils Dahlbäck:** Representation of Discourse-Cognitive and Computational Aspects, 1992, ISBN 91-7870-850-8.

No 265 **Ulf Nilsson:** Abstract Interpretations and Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs, 1992, ISBN 91-7870-858-3.

No 270 **Ralph Rönnquist:** Theory and Practice of Tense-bound Object References, 1992, ISBN 91-7870-873-7.

No 273 **Björn Fjellborg:** Pipeline Extraction for VLSI Data Path Synthesis, 1992, ISBN 91-7870-880-X.

No 276 **Staffan Bonnier:** A Formal Basis for Horn Clause Logic with External Polymorphic Functions, 1992, ISBN 91-7870-896-6.

No 277 **Kristian Sandahl:** Developing Knowledge Management Systems with an Active Expert Methodology, 1992, ISBN 91-7870-897-4.

No 281 **Christer Bäckström:** Computational Complexity of Reasoning about Plans, 1992, ISBN 91-7870-979-2.

No 292 **Mats Wirén:** Studies in Incremental Natural Language Analysis, 1992, ISBN 91-7871-027-8.

No 297 **Mariam Kamkar:** Interprocedural Dynamic Slicing with Applications to Debugging and Testing, 1993, ISBN 91-7871-065-0.

No 302 **Tingting Zhang:** A Study in Diagnosis Using Classification and Defaults, 1993, ISBN 91-7871-078-2

No 312 **Arne Jönsson:** Dialogue Management for Natural Language Interfaces - An Empirical Approach, 1993, ISBN 91-7871-110-X.

No 338 **Simin Nadjm-Tehrani:** Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification, 1994, ISBN 91-7871-237-8.

No 371 **Bengt Savén:** Business Models for Decision Support and Learning. A Study of Discrete-Event Manufacturing Simulation at Asea/ ABB 1968-1993, 1995, ISBN 91-7871-494-X.

No 375 **Ulf Söderman:** Conceptual Modelling of Mode Switching Physical Systems, 1995, ISBN 91-7871-516-4.

No 383 **Andreas Kågedal:** Exploiting Groundness in Logic Programs, 1995, ISBN 91-7871-538-5.

No 396 **George Fodor:** Ontological Control, Description, Identification and Recovery from Problematic Control Situations, 1995, ISBN 91-7871-603-9.

No 413 **Mikael Pettersson:** Compiling Natural Semantics, 1995, ISBN 91-7871-641-1.

No 414 **Xinli Gu:** RT Level Testability Improvement by Testability Analysis and Transformations, 1996, ISBN 91-7871-654-3.

No 416 **Hua Shu:** Distributed Default Reasoning, 1996, ISBN 91-7871-665-9.

No 429 **Jaime Villegas:** Simulation Supported Industrial Training from an Organisational Learning Perspective - Development and Evaluation of the SSIT Method, 1996, ISBN 91-7871-700-0.

No 431 **Peter Jonsson:** Studies in Action Planning: Algorithms and Complexity, 1996, ISBN 91-7871-704-3.

No 437 **Johan Boye:** Directional Types in Logic Programming, 1996, ISBN 91-7871-725-6.

No 439 **Cecilia Sjöberg:** Activities, Voices and Arenas: Participatory Design in Practice, 1996, ISBN 91-7871-728-0.

No 448 **Patrick Lambrix:** Part-Whole Reasoning in Description Logics, 1996, ISBN 91-7871-820-1.

No 452 **Kjell Orsborn:** On Extensible and Object-Relational Database Technology for Finite Element Analysis Applications, 1996, ISBN 91-7871-827-9.

No 459 **Olof Johansson:** Development Environments for Complex Product Models, 1996, ISBN 91-7871-855-4.

No 461 **Lena Strömbäck:** User-Defined Constructions in Unification-Based Formalisms, 1997, ISBN 91-7871-857-0.

No 462 **Lars Degerstedt:** Tabulation-based Logic Programming: A Multi-Level View of Query Answering, 1996, ISBN 91-7871-858-9.

No 475 **Fredrik Nilsson:** Strategi och ekonomisk styrning - En studie av hur ekonomiska styrsystem utformas och används efter företagsförvärv, 1997, ISBN 91-7871-914-3.

No 480 **Mikael Lindvall:** An Empirical Study of Requirements-Driven Impact Analysis in Object-Oriented Software Evolution, 1997, ISBN 91-7871-927-5.

No 485 **Göran Forslund:** Opinion-Based Systems: The Cooperative Perspective on Knowledge-Based Decision Support, 1997, ISBN 91-7871-938-0.

No 494 **Martin Sköld:** Active Database Management Systems for Monitoring and Control, 1997, ISBN 91-7219-002-7.

No 495 **Hans Olsén:** Automatic Verification of Petri Nets in a CLP framework, 1997, ISBN 91-7219-011-6.

No 498 **Thomas Drakengren:** Algorithms and Complexity for Temporal and Spatial Formalisms, 1997, ISBN 91-7219-019-1.

No 502 **Jakob Axelsson:** Analysis and Synthesis of Heterogeneous Real-Time Systems, 1997, ISBN 91-7219-035-3.

No 503 **Johan Ringström:** Compiler Generation for Data-Parallel Programming Languages from Two-Level Semantics Specifications, 1997, ISBN 91-7219-045-0.

No 512 **Anna Moberg:** Närhet och distans - Studier av kommunikationsmönster i satellitkontor och flexibla kontor, 1997, ISBN 91-7219-119-8.

No 520 **Mikael Ronström:** Design and Modelling of a Parallel Data Server for Telecom Applications, 1998, ISBN 91-7219-169-4.

No 522 **Niclas Ohlsson:** Towards Effective Fault Prevention - An Empirical Study in Software Engineering, 1998, ISBN 91-7219-176-7.

No 526 **Joachim Karlsson:** A Systematic Approach for Prioritizing Software Requirements, 1998, ISBN 91-7219-184-8.

No 530 **Henrik Nilsson:** Declarative Debugging for Lazy Functional Languages, 1998, ISBN 91-7219-197-x.

No 555 **Jonas Hallberg:** Timing Issues in High-Level Synthesis, 1998, ISBN 91-7219-369-7.

No 561 **Ling Lin:** Management of 1-D Sequence Data - From Discrete to Continuous, 1999, ISBN 91-7219-402-2.

No 563 **Eva L Ragnemalm:** Student Modelling based on Collaborative Dialogue with a Learning Companion, 1999, ISBN 91-7219-412-X.

No 567 **Jörgen Lindström:** Does Distance matter? On geographical dispersion in organisations, 1999, ISBN 91-7219-439-1.

No 582 **Vanja Josifovski:** Design, Implementation and Evaluation of a Distributed Mediator System for Data Integration, 1999, ISBN 91-7219-482-0.

No 589 **Rita Kovordányi:** Modeling and Simulating Inhibitory Mechanisms in Mental Image Reinterpretation - Towards Cooperative Human-Computer Creativity, 1999, ISBN 91-7219-506-1.

No 592 **Mikael Ericsson:** Supporting the Use of Design Knowledge - An Assessment of Commenting Agents, 1999, ISBN 91-7219-532-0.

No 593 **Lars Karlsson:** Actions, Interactions and Narratives, 1999, ISBN 91-7219-534-7.

No 594 **C. G. Mikael Johansson:** Social and Organizational Aspects of Requirements Engineering Methods - A practice-oriented approach, 1999, ISBN 91-7219-541-X.

No 595 **Jörgen Hansson:** Value-Driven Multi-Class Overload Management in Real-Time Database Systems, 1999, ISBN 91-7219-542-8.

No 596 **Niklas Hallberg:** Incorporating User Values in the Design of Information Systems and Services in the Public Sector: A Methods Approach, 1999, ISBN 91-7219-543-6.

No 597 **Vivian Vimarlund:** An Economic Perspective on the Analysis of Impacts of Information Technology: From Case Studies in Health-Care towards General Models and Theories, 1999, ISBN 91-7219-544-4.

No 598 **Johan Jenvald:** Methods and Tools in Computer-Supported Taskforce Training, 1999, ISBN 91-7219-547-9.

No 607 **Magnus Merkel:** Understanding and enhancing translation by parallel text processing, 1999, ISBN 91-7219-614-9.

No 611 **Silvia Coradeschi:** Anchoring symbols to sensory data, 1999, ISBN 91-7219-623-8.

No 613 **Man Lin:** Analysis and Synthesis of Reactive Systems: A Generic Layered Architecture Perspective, 1999, ISBN 91-7219-630-0.

No 618 **Jimmy Tjäder:** Systemimplementering i praktiken - En studie av logiker i fyra projekt, 1999, ISBN 91-7219-657-2.

No 627 **Vadim Engelson:** Tools for Design, Interactive Simulation, and Visualization of Object-Oriented Models in Scientific Computing, 2000, ISBN 91-7219-709-9.

No 637 **Esa Falkenroth:** Database Technology for Control and Simulation, 2000, ISBN 91-7219-766-8.

No 639 **Per-Arne Persson:** Bringing Power and Knowledge Together: Information Systems Design for Autonomy and Control in Command Work, 2000, ISBN 91-7219-796-X.

No 660 **Erik Larsson:** An Integrated System-Level Design for Testability Methodology, 2000, ISBN 91-7219-890-7.

No 688 **Marcus Bjäreland:** Model-based Execution Monitoring, 2001, ISBN 91-7373-016-5.

No 689 **Joakim Gustafsson:** Extending Temporal Action Logic, 2001, ISBN 91-7373-017-3.

No 720 **Carl-Johan Petri:** Organizational Information Provision - Managing Mandatory and Discretionary Use of Information Technology, 2001, ISBN-91-7373-126-9.

No 724 **Paul Scerri:** Designing Agents for Systems with Adjustable Autonomy, 2001, ISBN 91 7373 207 9.

No 725 **Tim Heyer:** Semantic Inspection of Software Artifacts: From Theory to Practice, 2001, ISBN 91 7373 208 7.

No 726 **Pär Carlshamre:** A Usability Perspective on Requirements Engineering - From Methodology to Product Development, 2001, ISBN 91 7373 212 5.

No 732 **Juha Takkinen:** From Information Management to Task Management in Electronic Mail, 2002, ISBN 91 7373 258 3.

No 745 **Johan Åberg:** Live Help Systems: An Approach to Intelligent Help for Web Information Systems, 2002, ISBN 91-7373-311-3.

No 746 **Rego Granlund:** Monitoring Distributed Teamwork Training, 2002, ISBN 91-7373-312-1.

No 757 **Henrik André-Jönsson:** Indexing Strategies for Time Series Data, 2002, ISBN 917373-346-6.

No 747 **Anneli Hagdahl:** Development of IT-supported Interorganisational Collaboration - A Case Study in the Swedish Public Sector, 2002, ISBN 91-7373-314-8.

No 749 **Sofie Pilemalm:** Information Technology for Non-Profit Organisations - Extended Participatory Design of an Information System for Trade Union Shop Stewards, 2002, ISBN 91-7373-318-0.

No 765 **Stefan Holmlid:** Adapting users: Towards a theory of use quality, 2002, ISBN 91-7373-397-0.

No 771 **Magnus Morin:** Multimedia Representations of Distributed Tactical Operations, 2002, ISBN 91-7373-421-7.

No 772 **Pawel Pietrzak:** A Type-Based Framework for Locating Errors in Constraint Logic Programs, 2002, ISBN 91-7373-422-5.

No 758 **Erik Berglund:** Library Communication Among Programmers Worldwide, 2002, ISBN 91-7373-349-0.

No 774 **Choong-ho Yi:** Modelling Object-Oriented Dynamic Systems Using a Logic-Based Framework, 2002, ISBN 91-7373-424-1.

No 779 **Mathias Broxvall:** A Study in the Computational Complexity of Temporal Reasoning, 2002, ISBN 91-7373-440-3.

No 793 **Asmus Pandikow:** A Generic Principle for Enabling Interoperability of Structured and Object-Oriented Analysis and Design Tools, 2002, ISBN 91-7373-479-9.

No 785 **Lars Hult:** Publika Informationstjänster. En studie av den Internetbaserade encyklopedins bruksegenskaper, 2003, ISBN 91-7373-461-6.

No 800 **Lars Taxén:** A Framework for the Coordination of Complex Systems´ Development, 2003, ISBN 91-7373-604-X

No 808 **Klas Gäre:** Tre perspektiv på förväntningar och förändringar i samband med införande av informationssystem, 2003, ISBN 91-7373-618-X.

No 821 **Mikael Kindborg:** Concurrent Comics - programming of social agents by children, 2003, ISBN 91-7373-651-1.

No 823 **Christina Ölvingson:** On Development of Information Systems with GIS Functionality in Public Health Informatics: A Requirements Engineering Approach, 2003, ISBN 91-7373-656-2.

No 828 **Tobias Ritzau:** Memory Efficient Hard Real-Time Garbage Collection, 2003, ISBN 91-7373-666-X.

No 833 **Paul Pop:** Analysis and Synthesis of Communication-Intensive Heterogeneous Real-Time Systems, 2003, ISBN 91-7373-683-X.

No 852 **Johan Moe:** Observing the Dynamic Behaviour of Large Distributed Systems to Improve Development and Testing – An Empirical Study in Software Engineering, 2003, ISBN 91-7373-779-8.

No 867 **Erik Herzog:** An Approach to Systems Engineering Tool Data Representation and Exchange, 2004, ISBN 91-7373-929-4.

No 872 **Aseel Berglund:** Augmenting the Remote Control: Studies in Complex Information Navigation for Digital TV, 2004, ISBN 91-7373-940-5.

No 869 **Jo Skåmedal:** Telecommuting's Implications on Travel and Travel Patterns, 2004, ISBN 91-7373-935-9.

No 870 **Linda Askenäs:** The Roles of IT - Studies of Organising when Implementing and Using Enterprise Systems, 2004, ISBN 91-7373-936-7.

No 874 **Annika Flycht-Eriksson:** Design and Use of Ontologies in Information-Providing Dialogue Systems, 2004, ISBN 91-7373-947-2.

No 873 **Peter Bunus:** Debugging Techniques for Equation-Based Languages, 2004, ISBN 91-7373-941-3.

No 876 **Jonas Mellin:** Resource-Predictable and Efficient Monitoring of Events, 2004, ISBN 91-7373-956-1.

No 883 **Magnus Bång:** Computing at the Speed of Paper: Ubiquitous Computing Environments for Healthcare Professionals, 2004, ISBN 91-7373-971-5

No 882 **Robert Eklund:** Disfluency in Swedish human-human and human-machine travel booking dialogues, 2004, ISBN 91-7373-966-9.

No 887 **Anders Lindström:** English and other Foreign Linguistic Elements in Spoken Swedish. Studies of Productive Processes and their Modelling using Finite-State Tools, 2004, ISBN 91-7373-981-2.

No 889 **Zhiping Wang:** Capacity-Constrained Production-inventory systems - Modelling and Analysis in both a traditional and an e-business context, 2004, ISBN 91-85295-08-6.

No 893 **Pernilla Qvarfordt:** Eyes on Multimodal Interaction, 2004, ISBN 91-85295-30-2.

No 910 **Magnus Kald:** In the Borderland between Strategy and Management Control - Theoretical Framework and Empirical Evidence, 2004, ISBN 91-85295-82-5.

No 918 **Jonas Lundberg:** Shaping Electronic News: Genre Perspectives on Interaction Design, 2004, ISBN 91-85297-14-3.

No 900 **Mattias Arvola:** Shades of use: The dynamics of interaction design for sociable use, 2004, ISBN 91-85295-42-6.

No 920 **Luis Alejandro Cortés:** Verification and Scheduling Techniques for Real-Time Embedded Systems, 2004, ISBN 91-85297-21-6.

No 929 **Diana Szentivanyi:** Performance Studies of Fault-Tolerant Middleware, 2005, ISBN 91-85297-58-5.

No 933 **Mikael Cäker:** Management Accounting as Constructing and Opposing Customer Focus: Three Case Studies on Management Accounting and Customer Relations, 2005, ISBN 91-85297-64-X.

No 937 **Jonas Kvarnström:** TALplanner and Other Extensions to Temporal Action Logic, 2005, ISBN 91-85297-75-5.

No 938 **Bourhane Kadmiry:** Fuzzy Gain-Scheduled Visual Servoing for Unmanned Helicopter, 2005, ISBN 91-85297-76-3.

No 945 **Gert Jervan:** Hybrid Built-In Self-Test and Test Generation Techniques for Digital Systems, 2005, ISBN: 91-85297-97-6.

No 946 **Anders Arpteg:** Intelligent Semi-Structured Information Extraction, 2005, ISBN 91-85297-98-4.

No 947 **Ola Angelsmark:** Constructing Algorithms for Constraint Satisfaction and Related Problems - Methods and Applications, 2005, ISBN 91-85297-99-2.

No 963 **Calin Curescu:** Utility-based Optimisation of Resource Allocation for Wireless Networks, 2005, ISBN 91-85457-07-8.

No 972 **Björn Johansson:** Joint Control in Dynamic Situations, 2005, ISBN 91-85457-31-0.

No 974 **Dan Lawesson:** An Approach to Diagnosability Analysis for Interacting Finite State Systems, 2005, ISBN 91-85457-39-6.

No 979 **Claudiu Duma:** Security and Trust Mechanisms for Groups in Distributed Services, 2005, ISBN 91-85457-54-X.

No 983 **Sorin Manolache:** Analysis and Optimisation of Real-Time Systems with Stochastic Behaviour, 2005, ISBN 91-85457-60-4.

No 986 **Yuxiao Zhao:** Standards-Based Application Integration for Business-to-Business Communications, 2005, ISBN 91-85457-66-3.

No 1004 **Patrik Haslum:** Admissible Heuristics for Automated Planning, 2006, ISBN 91-85497-28-2.

No 1005 **Aleksandra Tešanovic:** Developing Reusable and Reconfigurable Real-Time Software using Aspects and Components, 2006, ISBN 91-85497-29-0.

No 1008 **David Dinka:** Role, Identity and Work: Extending the design and development agenda, 2006, ISBN 91-85497-42-8.

No 1009 **Iakov Nakhimovski:** Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analysis, 2006, ISBN 91-85497-43-X.

No 1013 **Wilhelm Dahllöf:** Exact Algorithms for Exact Satisfiability Problems, 2006, ISBN 91-85523-97-6.

No 1016 **Levon Saldamli:** PDEModelica - A High-Level Language for Modeling with Partial Differential Equations, 2006, ISBN 91-85523-84-4.

No 1017 **Daniel Karlsson:** Verification of Component-based Embedded System Designs, 2006, ISBN 91-85523-79-8

No 1018 **Ioan Chisalita:** Communication and Networking Techniques for Traffic Safety Systems, 2006, ISBN 91-85523-77-1.

No 1019 **Tarja Susi:** The Puzzle of Social Activity - The Significance of Tools in Cognition and Cooperation, 2006, ISBN 91-85523-71-2.

No 1021 **Andrzej Bednarski:** Integrated Optimal Code Generation for Digital Signal Processors, 2006, ISBN 91-85523-69-0.

No 1022 **Peter Aronsson:** Automatic Parallelization of Equation-Based Simulation Programs, 2006, ISBN 91-85523-68-2.

No 1030 **Robert Nilsson:** A Mutation-based Framework for Automated Testing of Timeliness, 2006, ISBN 91-85523-35-6.

No 1034 **Jon Edvardsson:** Techniques for Automatic Generation of Tests from Programs and Specifications, 2006, ISBN 91-85523-31-3.

No 1035 **Vaida Jakoniene:** Integration of Biological Data, 2006, ISBN 91-85523-28-3.

No 1045 **Genevieve Gorrell:** Generalized Hebbian Algorithms for Dimensionality Reduction in Natural Language Processing, 2006, ISBN 91-85643-88-2.

No 1051 **Yu-Hsing Huang:** Having a New Pair of Glasses - Applying Systemic Accident Models on Road Safety, 2006, ISBN 91-85643-64-5.

No 1054 **Åsa Hedenskog:** Perceive those things which cannot be seen - A Cognitive Systems Engineering perspective on requirements management, 2006, ISBN 91-85643-57-2.

No 1061 **Cécile Åberg:** An Evaluation Platform for Semantic Web Technology, 2007, ISBN 91-85643-31-9.

No 1073 **Mats Grindal:** Handling Combinatorial Explosion in Software Testing, 2007, ISBN 978-91-85715-74-9.

No 1075 **Almut Herzog:** Usable Security Policies for Runtime Environments, 2007, ISBN 978-91-85715-65-7.

No 1079 **Magnus Wahlström:** Algorithms, measures, and upper bounds for Satisfiability and related problems, 2007, ISBN 978-91-85715-55-8.

No 1083 **Jesper Andersson:** Dynamic Software Architectures, 2007, ISBN 978-91-85715-46-6.

No 1086 **Ulf Johansson:** Obtaining Accurate and Comprehensible Data Mining Models - An Evolutionary Approach, 2007, ISBN 978-91-85715-34-3.

No 1089 **Traian Pop:** Analysis and Optimisation of Distributed Embedded Systems with Heterogeneous Scheduling Policies, 2007, ISBN 978-91-85715-27-5.

No 1091 **Gustav Nordh:** Complexity Dichotomies for CSP-related Problems, 2007, ISBN 978-91-85715-20-6.

No 1106 **Per Ola Kristensson:** Discrete and Continuous Shape Writing for Text Entry and Control, 2007, ISBN 978-91-85831-77-7.

No 1110 **He Tan:** Aligning Biomedical Ontologies, 2007, ISBN 978-91-85831-56-2.

No 1112 **Jessica Lindblom:** Minding the body - Interacting socially through embodied action, 2007, ISBN 978-91-85831-48-7.

No 1113 **Pontus Wärnestål:** Dialogue Behavior Management in Conversational Recommender Systems, 2007, ISBN 978-91-85831-47-0.

No 1120 **Thomas Gustafsson:** Management of Real-Time Data Consistency and Transient Overloads in Embedded Systems, 2007, ISBN 978-91-85831-33-3.

No 1127 **Alexandru Andrei:** Energy Efficient and Predictable Design of Real-time Embedded Systems, 2007, ISBN 978-91-85831-06-7.

No 1139 **Per Wikberg:** Eliciting Knowledge from Experts in Modeling of Complex Systems: Managing Variation and Interactions, 2007, ISBN 978-91-85895-66-3.

No 1143 **Mehdi Amirijoo:** QoS Control of Real-Time Data Services under Uncertain Workload, 2007, ISBN 978-91-85895-49-6.

No 1150 **Sanny Syberfeldt:** Optimistic Replication with Forward Conflict Resolution in Distributed Real-Time Databases, 2007, ISBN 978-91-85895-27-4.

No 1155 **Beatrice Alenljung:** Envisioning a Future Decision Support System for Requirements Engineering - A Holistic and Human-centred Perspective, 2008, ISBN 978-91-85895-11-3.

No 1156 **Artur Wilk:** Types for XML with Application to Xcerpt, 2008, ISBN 978-91-85895-08-3.

No 1183 **Adrian Pop:** Integrated Model-Driven Development Environments for Equation-Based Object-Oriented Languages, 2008, ISBN 978-91-7393-895-2.

No 1185 **Jörgen Skågeby:** Gifting Technologies - Ethnographic Studies of End-users and Social Media Sharing, 2008, ISBN 978-91-7393-892-1.

No 1187 **Imad-Eldin Ali Abugessaisa:** Analytical tools and information-sharing methods supporting road safety organizations, 2008, ISBN 978-91-7393-887-7.

No 1204 **H. Joe Steinhauer:** A Representation Scheme for Description and Reconstruction of Object Configurations Based on Qualitative Relations, 2008, ISBN 978-91-7393-823-5.

No 1222 **Anders Larsson:** Test Optimization for Core-based System-on-Chip, 2008, ISBN 978-91-7393-768-9.

No 1238 **Andreas Borg:** Processes and Models for Capacity Requirements in Telecommunication Systems, 2009, ISBN 978-91-7393-700-9.

No 1240 **Fredrik Heintz:** DyKnow: A Stream-Based Knowledge Processing Middleware Framework, 2009, ISBN 978-91-7393-696-5.

No 1241 **Birgitta Lindström:** Testability of Dynamic Real-Time Systems, 2009, ISBN 978-91-7393-695-8.

No 1244 **Eva Blomqvist:** Semi-automatic Ontology Construction based on Patterns, 2009, ISBN 978-91-7393-683-5.

No 1249 **Rogier Woltjer:** Functional Modeling of Constraint Management in Aviation Safety and Command and Control, 2009, ISBN 978-91-7393-659-0.

No 1260 **Gianpaolo Conte:** Vision-Based Localization and Guidance for Unmanned Aerial Vehicles, 2009, ISBN 978-91-7393-603-3.

No 1262 **AnnMarie Ericsson:** Enabling Tool Support for Formal Analysis of ECA Rules, 2009, ISBN 978-91-7393-598-2.

No 1266 **Jiri Trnka:** Exploring Tactical Command and Control: A Role-Playing Simulation Approach, 2009, ISBN 978-91-7393-571-5.

No 1268 **Bahlol Rahimi:** Supporting Collaborative Work through ICT - How End-users Think of and Adopt Integrated Health Information Systems, 2009, ISBN 978-91-7393-550-0.

No 1274 **Fredrik Kuivinen:** Algorithms and Hardness Results for Some Valued CSPs, 2009, ISBN 978-91-7393-525-8.

No 1281 **Gunnar Mathiason:** Virtual Full Replication for Scalable Distributed Real-Time Databases, 2009, ISBN 978-91-7393-503-6.

No 1290 **Viacheslav Izosimov:** Scheduling and Optimization of Fault-Tolerant Distributed Embedded Systems, 2009, ISBN 978-91-7393-482-4.

No 1294 **Johan Thapper:** Aspects of a Constraint Optimisation Problem, 2010, ISBN 978-91-7393-464-0.

No 1306 **Susanna Nilsson:** Augmentation in the Wild: User Centered Development and Evaluation of Augmented Reality Applications, 2010, ISBN 978-91-7393-416-9.

No 1313 **Christer Thörn:** On the Quality of Feature Models, 2010, ISBN 978-91-7393-394-0.

No 1321 **Zhiyuan He:** Temperature Aware and Defect-Probability Driven Test Scheduling for System-on-Chip, 2010, ISBN 978-91-7393-378-0.

No 1333 **David Broman:** Meta-Languages and Semantics for Equation-Based Modeling and Simulation, 2010, ISBN 978-91-7393-335-3.

No 1337 **Alexander Siemers:** Contributions to Modelling and Visualisation of Multibody Systems Simulations with Detailed Contact Analysis, 2010, ISBN 978-91-7393-317-9.

No 1354 **Mikael Asplund:** Disconnected Discoveries: Availability Studies in Partitioned Networks, 2010, ISBN 978-91-7393-278-3.

No 1359 **Jana Rambusch**: Mind Games Extended: Understanding Gameplay as Situated Activity, 2010, ISBN 978-91-7393-252-3.

No 1373 **Sonia Sangari**: Head Movement Correlates to Focus Assignment in Swedish,2011,ISBN 978-91-7393-154-0.

No 1374 **Jan-Erik Källhammer**: Using False Alarms when Developing Automotive Active Safety Systems, 2011, ISBN 978-91-7393-153-3.

No 1375 **Mattias Eriksson**: Integrated Code Generation, 2011, ISBN 978-91-7393-147-2.

No 1381 **Ola Leifler**: Affordances and Constraints of Intelligent Decision Support for Military Command and Control – Three Case Studies of Support Systems, 2011, ISBN 978-91-7393-133-5.

No 1386 **Soheil Samii**: Quality-Driven Synthesis and Optimization of Embedded Control Systems, 2011, ISBN 978-91-7393-102-1.

No 1419 **Erik Kuiper**: Geographic Routing in Intermittently-connected Mobile Ad Hoc Networks: Algorithms and Performance Models, 2012, ISBN 978-91-7519-981-8.

No 1451 **Sara Stymne**: Text Harmonization Strategies for Phrase-Based Statistical Machine Translation, 2012, ISBN 978-91-7519-887-3.

No 1455 **Alberto Montebelli**: Modeling the Role of Energy Management in Embodied Cognition, 2012, ISBN 978-91-7519-882-8.

No 1465 **Mohammad Saifullah**: Biologically-Based Interactive Neural Network Models for Visual Attention and Object Recognition, 2012, ISBN 978-91-7519-838-5.

No 1490 **Tomas Bengtsson**: Testing and Logic Optimization Techniques for Systems on Chip, 2012, ISBN 978-91-7519-742-5.

No 1481 **David Byers**: Improving Software Security by Preventing Known Vulnerabilities, 2012, ISBN 978-91-7519-784-5.

No 1496 **Tommy Färnqvist**: Exploiting Structure in CSP-related Problems, 2013, ISBN 978-91-7519-711-1.

No 1503 **John Wilander**: Contributions to Specification, Implementation, and Execution of Secure Software, 2013, ISBN 978-91-7519-681-7.

No 1506 **Magnus Ingmarsson**: Creating and Enabling the Useful Service Discovery Experience, 2013, ISBN 978-91-7519-662-6.

No 1547 **Wladimir Schamai**: Model-Based Verification of Dynamic System Behavior against Requirements: Method, Language, and Tool, 2013, ISBN 978-91-7519-505-6.

No 1551 **Henrik Svensson**: Simulations, 2013, ISBN 978-91-7519-491-2.

No 1559 **Sergiu Rafiliu**: Stability of Adaptive Distributed Real-Time Systems with Dynamic Resource Management, 2013, ISBN 978-91-7519-471-4.

No 1581 **Usman Dastgeer**: Performance-aware Component Composition for GPU-based Systems, 2014, ISBN 978-91-7519-383-0.

No 1602 **Cai Li**: Reinforcement Learning of Locomotion based on Central Pattern Generators, 2014, ISBN 978-91-7519-313-7.

No 1652 **Roland Samlaus**: An Integrated Development Environment with Enhanced Domain-Specific

Interactive Model Validation, 2015, ISBN 978-91-7519-090-7.

No 1663 **Hannes Uppman**: On Some Combinatorial Optimization Problems: Algorithms and Complexity, 2015, ISBN 978-91-7519-072-3.

No 1664 **Martin Sjölund**: Tools and Methods for Analysis, Debugging, and Performance Improvement of Equation-Based Models, 2015, ISBN 978-91-7519-071-6.

No 1666 **Kristian Stavåker**: Contributions to Simulation of Modelica Models on Data-Parallel Multi-Core Architectures, 2015, ISBN 978-91-7519-068-6.

No 1680 **Adrian Lifa**: Hardware/ Software Codesign of Embedded Systems with Reconfigurable and Heterogeneous Platforms, 2015, ISBN 978-91-7519-040-2.

No 1685 **Bogdan Tanasa**: Timing Analysis of Distributed Embedded Systems with Stochastic Workload and Reliability Constraints, 2015, ISBN 978-91-7519-022-8.

No 1691 **Håkan Warnquist**: Troubleshooting Trucks – Automated Planning and Diagnosis, 2015, ISBN 978-91-7685-993-3.

No 1702 **Nima Aghaee**: Thermal Issues in Testing of Advanced Systems on Chip, 2015, ISBN 978-91-7685-949-0.

No 1715 **Maria Vasilevskaya**: Security in Embedded Systems: A Model-Based Approach with Risk Metrics, 2015, ISBN 978-91-7685-917-9.

No 1729 **Ke Jiang**: Security-Driven Design of Real-Time Embedded System, 2016, ISBN 978-91-7685-884-4.

No 1733 **Victor Lagerkvist**: Strong Partial Clones and the Complexity of Constraint Satisfaction Problems: Limitations and Applications, 2016, ISBN 978-91-7685-856-1.

No 1734 **Chandan Roy**: An Informed System Development Approach to Tropical Cyclone Track and Intensity Forecasting, 2016, ISBN 978-91-7685-854-7.

No 1746 **Amir Aminifar**: Analysis, Design, and Optimization of Embedded Control Systems, 2016, ISBN 978-91-7685-826-4.

No 1747 **Ekhiotz Vergara**: Energy Modelling and Fairness for Efficient Mobile Communication, 2016, ISBN 978-91-7685-822-6.

No 1748 **Dag Sonntag**: Chain Graphs – Interpretations, Expressiveness and Learning Algorithms, 2016, ISBN 978-91-7685-818-9.

Linköping Studies in Arts and Science

No 504 **Ing-Marie Jonsson:** Social and Emotional Characteristics of Speech-based In-Vehicle Information Systems: Impact on Attitude and Driving Behaviour, 2009, ISBN 978-91-7393-478-7.

No 586 **Fabian Segelström:** Stakeholder Engagement for Service Design: How service designers identify and communicate insights, 2013, ISBN 978-91-7519-554-4.

No 618 **Johan Blomkvist:** Representing Future Situations of Service: Prototyping in Service Design, 2014, ISBN 978-91-7519-343-4.

No 620 **Marcus Mast:** Human-Robot Interaction for Semi-Autonomous Assistive Robots, 2014, ISBN 978-91-7519-319-9.

Linköping Studies in Statistics

No 9 **Davood Shahsavani:** Computer Experiments Designed to Explore and Approximate Complex Deterministic Models, 2008, ISBN 978-91-7393-976-8.

No 10 **Karl Wahlin:** Roadmap for Trend Detection and Assessment of Data Quality, 2008, ISBN 978-91-7393-792-4.

No 11 **Oleg Sysoev:** Monotonic regression for large multivariate datasets, 2010, ISBN 978-91-7393-412-1.

No 13 **Agné Burauskaite-Harju:** Characterizing Temporal Change and Inter-Site Correlations in Daily and Subdaily Precipitation Extremes, 2011, ISBN 978-91-7393-110-6.

Linköping Studies in Information Science

No 1 **Karin Axelsson:** Metodisk systemstrukturering- att skapa samstämmighet mellan informationssystemarkitektur och verksamhet, 1998. ISBN-9172-19-296-8.

No 2 **Stefan Cronholm:** Metodverktyg och användbarhet - en studie av datorstödd metodbaserad systemutveckling, 1998, ISBN-9172-19-299-2.

No 3 **Anders Avdic:** Användare och utvecklare - om anveckling med kalkylprogram, 1999. ISBN-91-7219-606-8.

No 4 **Owen Eriksson:** Kommunikationskvalitet hos informationssystem och affärsprocesser, 2000, ISBN 91-7219-811-7.

No 5 **Mikael Lind:** Från system till process - kriterier för processbestämning vid verksamhetsanalys, 2001, ISBN 91-7373-067-X.

No 6 **Ulf Melin:** Koordination och informationssystem i företag och nätverk, 2002, ISBN 91-7373-278-8.

No 7 **Pär J. Ågerfalk:** Information Systems Actability - Understanding Information Technology as a Tool for Business Action and Communication, 2003, ISBN 91-7373-628-7.

No 8 **Ulf Seigerroth:** Att förstå och förändra systemutvecklingsverksamheter - en taxonomi för metautveckling, 2003, ISBN91-7373-736-4.

No 9 **Karin Hedström:** Spår av datoriseringens värden – Effekter av IT i äldreomsorg, 2004, ISBN 91-7373-963-4.

No 10 **Ewa Braf:** Knowledge Demanded for Action - Studies on Knowledge Mediation in Organisations, 2004, ISBN 91-85295-47-7.

No 11 **Fredrik Karlsson:** Method Configuration method and computerized tool support, 2005, ISBN 91-85297-48-8.

No 12 **Malin Nordström:** Styrbar systemförvaltning - Att organisera systemförvaltningsverksamhet med hjälp av effektiva förvaltningsobjekt, 2005, ISBN 91-85297-60-7.

No 13 **Stefan Holgersson:** Yrke: POLIS - Yrkeskunskap, motivation, IT-system och andra förutsättningar för polisarbete, 2005, ISBN 91-85299-43-X.

No 14 **Benneth Christiansson, Marie-Therese Christiansson:** Mötet mellan process och komponent - mot ett ramverk för en verksamhetsnära kravspecifikation vid anskaffning av komponentbaserade informationssystem, 2006, ISBN 91-85643-22-X.