Qualitative Possibilistic Graphical Models: From Independence to Propagation Algorithms

Nahla Ben Amor

January 5, 2012

Abstract

This thesis proposes a study of qualitative possibilistic independence relations and explores propagation algorithms for possibilistic causal networks. We first show the existence of two forms of independence relations: causal and decompositional. Then, we propose new forms of independence relations based on plausibility relations underlying possibility distributions. The proposed relations are compared with those already known in possibilistic framework. The obtained results are used in defining possibilistic counterparts of probabilistic Bayesian networks. In fact, two kinds of networks are studied, depending if the possibilistic conditioning is based on the minimum or the product operator. We first propose an adaptation of probabilistic propagation algorithms. This adaptation shows that when we use the product form of conditioning, we get possibilistic networks similar to probabilistic ones sharing the same features and having the same theoretical and practical results since conditioning is defined in the same way in the two frameworks. However, this is not the case with min-based networks. The particular properties of the minimum operator, such as the idempotency, lead us to explore a new anytime propagation approach for min-based networks which avoids the transformation of the initial network into a junction tree, known to be a hard problem. Experimentation results show the merits of our new propagation algorithm.

Résumé

Cette thèse propose une étude des relations d'indépendance possibilistes qualitatives et explore les algorithmes de propagation dans les réseaux causaux possibilistes. D'abord, nous montrons l'existence de deux types de relations d'indépendance: causales et décompositionelles. Ensuite, nous proposons de nouvelles définitions uniquement basées sur les relations de plausibilités sous-jacentes aux distributions de possibilités. Les relations d'indépendance proposées sont comparées à celles déjà existantes dans la théorie des possibilités. Ces résultats sur l'indépendance sont utilisés afin de proposer une contrepartie possibiliste des réseaux Bayésiens. En effet, deux types de réseaux possibilistes sont étudiés selon que le conditionnement est basé sur l'opérateur minimum ou sur l'opérateur produit. En premier lieu, nous proposons une adaptation possibiliste des algorithmes de propagation développés dans le cadre de la théorie des probabilités. Cette adaptation montre que les réseaux causaux basés sur l'opérateur produit ont des propriétés très similaires aux réseaux Bayésiens, ce qui n'est pas le cas lorsque le conditionnement est basé sur l'opérateur minimum. Les propriétés particulières de l'opérateur minimum telles que l'idempotence nous ont poussé à développer un nouvel algorithme *anytime* pour les réseaux causaux possibilistes basés sur le conditionnement ordinal. Le but est d'éviter la transformation coûteuse du graphe initial en un arbre de jonction. L'étude expérimentale montre les apports de notre nouvel algorithme de propagation.

Acknowledgments

I would like to thank my advisor Professor Khaled Mellouli who guided my first steps in the world of research. Thank you for having accepted to be the director of this thesis and for your insightful advices and continuous encouragements.

I owe so much to my co-advisor Salem Benferhat in the fulfillment of this thesis. First as he guided my initial ideas, then by his constant encouragement to publish the current state of my work and finally during the elaboration of this thesis which he supervised at all stages. Thank you for your trust and support.

I would like to express my considereation to Didier Dubois and Henri Prade, who received me several times in IRIT laboratory. I feel really fortunate to have met them and would like to express my gratitude for their valuable discussions and comments.

My thanks go to Didier Dubois and Foued Ben Abdelaziz for the interest they have made to this work by accepting to be my examiners.

I also would like to thank Parakash Shenoy for his valuable comments on our propagation algorithm.

Special thanks go to all the members of IRIT laboratory for their friendly support.

I am also thankful to all persons from *Institut des Hautes Etudes Commerciales de Carthage*, *Institut National des Sciences Appliquées et des Technologies* and *Institut Supérieur de Gestion de Tunis* who help me during this work.

A special wink to all the members of my laboratory LARODEC and especially Zied Elouedi, Fédia Khalfallah and Lamia Enneifer for all the good and bad times we get together.

Thanks to all the people who have been involved in this work and especially to Nizar Souissi for helping me in programming the interface of my software.

Contents

List of Figures

List of Tables

General Introduction

Artificial Intelligence aims to model human reasoning in order to help decision makers in their tasks. The development of expert systems is one of the most famous applications in this domain. Nevertheless, the first expert systems are unable to manipulate correctly incontrollable variables, due to the imprecise and uncertain information characterizing the real world.

Graphical models are important tools proposed for an efficient representation and analysis of uncertain information commonly used by an increasing number of researchers from different domains as industry [76], space [79], and medicine [77, 83]. Well-known graphical models are probabilistic *Bayesian networks* [84, 94, 103], *decision trees* [106], *Influence diagrams* [81, 107, 108] and *Valuation Based Systems* (VBS) [111, 112].

The success of graphical representations is due to their capacity of representing and handling independence relationships, which have been proved to be crucial for an efficient management and storage of the information. Moreover, graphical models meet our requirements of explicitness and clarity since graphs topologies (nodes for variables, edges for local dependencies among variables) allow a local representation and reasoning easily supported by human mind.

Most of proposed graphical models refer to probability theory. In particular, probabilistic Bayesian networks have been largely developed and used in real world applications. However, this framework is only appropriate when all numerical data are available, which is not always possible. Indeed, there are some situations, like the case of total ignorance, which are not well handled and which can make the probabilistic reasoning unsound. This is particularly true in probabilistic Bayesian networks when missing data do not allow any valid treatment.

Several non-classical theories of uncertainty have been proposed in order to deal with uncertain and imprecise data such as evidence theory [110, 117, 118], Lehmann's ranked model [96], plausibility relations [75], Spohn's ordinal conditional functions [119, 120] and possibility

theory [54, 60, 130] issued from fuzzy sets theory [86, 128].

Possibility theory offers a natural and simple model to handle uncertain information. It is an appropriate framework for experts to express their opinions about uncertainty *numerically* using possibility degrees or *qualitatively* using total pre-order on the universe of discourse. This theory has been used in different areas such as *default reasoning* [19], *qualitative decision* [59], *data fusion* [20, 57] and *diagnostic* [44]. Possibility theory has also a logical-based reasoning called *possibilistic logic* [52, 91]. This logic is an extension of classical logic where propositional or first-order formulas are weighted by lower bounds and necessity (or possibility) measures.

The aim of this thesis is to develop graphical models for reasoning with qualitative uncertain information. We are in particular interested in *qualitative possibilistic causal networks*, which are possibilistic counterparts of probabilistic Bayesian networks. Such networks are useful for experts which are unable to provide exact numerical values to quantify different links between variables.

Qualitative means here that we more focus on the pre-ordering on events (called plausibility relation) induced by possibility distributions rather than on the numerical values (possibility degrees) associated with events.

Qualitative causal possibilistic networks present a new promising area of research. Indeed, they offer a natural way to treat non binary variables. Moreover, they handle structured knowledge using independence relations between variables and allow local computations. These treatments are not supported by possibilistic logic.

In possibility theory, there are two different ways to define the counterpart of causal probabilistic (Bayesian) networks. This is due to the existence of two definitions of possibilistic conditioning [38, 39, 54, 78]: *product-based* conditioning and *min-based* conditioning. When we use the min-based conditioning, we speak about *qualitative possibilistic networks*.

Existing works on possibilistic graphical models are either a direct adaptation of probabilistic approach (without any care to knowledge representation) or a way to perform learning from imprecise data [69, 24]. Regarding the possibilistic propagation, there are only few works. We can mention the possibilistic propagation in hypergraphs proposed by Dubois and Prade [55], the adaptation of Pearl's algorithm by Fonck [63]. Gebhardt, Kruse and Borgelt have

proposed a software for possibilistic propagation in undirected possibilistic networks, called POSSINFER [22, 68, 69]. Shenoy has proposed a propagation algorithm in *Valuation Based Systems* (VBS) [111, 113, 114, 115]. VBS are general tools since valuations maybe particularized to a possibility distribution, a probability distribution, or a belief function. However, the possibilistic version of VBS does not include the qualitative possibilistic networks [112].

In order to develop propagation algorithms for *qualitative possibilistic graphical models*, it is important to define qualitative counterpart of stochastic independence which is not as well defined as in probability theory. Thus, the first part (Part I) of this thesis provides an analysis of the notion of *qualitative possibilistic independence.*

There has been a considerable interest in the last few years for discussing independence in various representation frameworks [12, 31, 37, 38, 39, 40, 41, 62, 121]. Conditional independence relations between variables play an important role in the handling of uncertain information. From an operational point of view, two forms of independence can be distinguished [2, 4]:
Decompositional independence which ensures the decomposition of a joint distribution pertaining to tuples of variables into local distributions on smaller subsets of variables. A reasoning machinery can then work at a local level without losing any information.
Causal independence for expressing the lack of causality between variables. This form of independence is always characterized in semantic terms. Roughly speaking, a variable (or set of variables) is said to have no influence on another variable (or set of variables) if our belief in the value of the latter does not change when learning something about the value of the former.

These two kinds of independence are not necessarily exclusive. Ideally, a good definition of independence expresses both the lack of causality (so it can be easily expressed by experts), and is useful for computations. In the probabilistic framework causal and decompositional independence relations are equivalent, which is not the case in possibility theory. In this thesis we investigate several definitions of independence based on possibility distributions and on the ranking induced by these distributions. In addition, we provide a comparative study between these independence relations and study their graphoid properties.

After analyzing different forms of qualitative independence relations. Part II of this thesis proposes an adaptation of probabilistic propagation algorithms. When we use the product form of conditioning, we get possibilistic networks similar to probabilistic ones sharing the

same features and having the same theoretical and practical results since conditioning is defined in the same way in the two frameworks. However, this is not the case with min-based networks since they do not satisfy the so-called *coherence* property pointed out by Fonck [63]. Indeed, it may happen that the joint distribution associated with the possibilistic graph do not recover the initial data provided by experts. Nevertheless, and as we will show, the coherence problem should not be seen as a drawback either in recovering independence relations or in the propagation process [9].

Our study of the different adaptations of probabilistic propagation algorithms, shows that the min-based propagation can be seen differently from the classical approach since the minimum operator has different properties from the product operator (used in both Bayesian and product-based networks) like the *idempotency* property. Therefore, we propose in the last part (Part III) of the thesis, a new propagation algorithm for min-based possibilistic networks which is not a direct adaptation of classical approach. In particular, we will avoid the transformation of the initial network into a junction tree known to be a hard problem [30].

The proposed algorithm is an anytime algorithm [8, 10]. It is composed of several steps such that the longer the algorithm runs, the closer to the exact marginals we get. The first step consists in transforming the initial possibilistic graph into an equivalent undirected graph, called here for simplicity *moral graph*, where each node (called cluster) contains a variable from the initial graph and its parents. The clusters are quantified by local joint distributions instead of the initial conditional ones.

In the second step, several stability procedures are used in order to guarantee that joint distributions on a given cluster are in agreement with those of its adjacent clusters. We start by a *simple stability procedure* which ensures that any cluster agrees with each of its adjacent clusters on the distributions defined on common variables. This procedure does not guarantee exact marginals. Thus, we propose to improve it by using a *multiple nodes stability procedure* which ensures that any cluster agrees on the distributions defined on common variables computed from 2, 3,.., n adjacent clusters. We will consider the case where nodes are all neighbors and also the cases where nodes are restricted to parents, children and parents with children. We also develop a *best multiple nodes stability procedure* which ensures that only best instances in the distribution of each cluster agree with the best instances in the distribution computed from several of its adjacent clusters.

Finally, in the last step, we propose two *consistency procedures* providing exact marginals. The first one is based on adding some links in the moral graph, while the second procedure is based on constructing best global instances.

This thesis is organized into three parts:

Qualitative Possibilistic Independence. In this part, Chapter 1 presents non-classical theories of uncertainty and essentially possibility theory. Chapter 2 proposes a qualitative uncertainty framework where uncertainty is represented by total pre-orders on the universe of discourse. Chapter 3 defines the notion of qualitative independence and compares it with basic existing independence relations in possibility theory. This chapter also provides a software allowing to test independence relations satisfied by any possibility distribution or plausibility relation.

Possibilistic Adaptation of Probabilistic Causal Networks. In this part, Chapter 4 presents probabilistic Bayesian networks and their propagation algorithms. Chapter 5 develops a possibilistic counterpart of probabilistic Bayesian networks based on the *minimum* and the *product* operators and discusses the *coherence* problem. Chapter 6 develops a possibilistic adaptation of exact probabilistic propagation algorithms for product and min based possibilistic networks. More precisely, we propose an adaptation of the centralized version of Pearl's algorithm [87, 103, 105] and of the probabilistic propagation in junction trees [84].

New Approach in Possibilistic Propagation. In this part, Chapter 7 proposes an anytime propagation algorithm for min-based possibilistic networks which avoid a direct adaptation of probabilistic propagation algorithms. Chapter 8 provides experimental results showing the merits of our algorithm comparing with classical approaches. Moreover, it proposes a Possibilistic Networks Toolbox (**PNT**) implemented with Matlab 6.0 allowing the propagation in both min-based and product-based possibilistic networks.

Notations and definitions

We first give some notations and definitions used in this thesis.
Let $V = \{A_1, A_2, ..., A_N\}$ be a set of variables,

- $D_A = \{a_1, ..., a_n\}$ denotes the supposedly finite domain associated with the variable A,
- a_i denotes any instance of A_i,
- $X, Y, Z, ...$, denote subsets of variables from V,
- $D_X = \times_{A_i \in X} D_{A_i}$ denotes the cartesian product of domains of variables in X,
- x denotes any instance of X, if $X = \{A_1, ..., A_n\}$ then $x = (a_1, ..., a_n)$,
- $\Omega = \times_{A_i \in V} D_{A_i}$ denotes the universe of discourse, which is the cartesian product of all variable domains in V,
- Each element $\omega \in \Omega$ is called an *interpretation*, a possible *world* or a *state* of Ω. Depending on the context, we use one of the following notations:
  - either tuples: $\omega = (a_1, ..., a_N)$
  - or conjunctions: $\omega = a_1 \wedge ... \wedge a_N$, then $\omega[A_i] = a_i$.
- ϕ, ψ, φ denote the subclasses of Ω (called propositions or events) and $\neg\phi$ denotes the complementary set of ϕ i.e. $\neg\phi = \Omega - \phi$,
- $\phi \wedge \psi$ (resp. $\phi \vee \psi$) denotes the intersection (resp. the union) of ϕ and ψ,
- $[a_i] = \{\omega = (a_1 \wedge ... \wedge a_N) : A_i = a_i\}$ denotes the set of states whose i^{th} component is a_i,
- $\forall x \in D_X, [x] = \{\omega = (a_1 \wedge ... \wedge a_N) : \forall A_i \in X, A_i = a_i\}$ denotes the set of states whose restrictions to variables in X is x.

  When there is no ambiguity, we use x instead of $[x]$ and $x \wedge y$ (resp. $x \vee y$) instead of $[x] \wedge [y]$ (resp. $[x] \vee [y]$).

Part I

Qualitative Possibilistic Independence

Introduction Part I

Conditional independence relations between variables play an important role in the handling of uncertain information. From an operational point of view, two forms of independence can be distinguished.

Decompositional independence which ensures the decomposition of a joint distribution pertaining to tuples of variables into local distributions on smaller subsets of variables. A reasoning machinery can then work at a local level without losing any information.
Causal independence for expressing the lack of causality between variables. This form of independence is always characterized in semantic terms. Roughly speaking, a variable (or set of variables) is said to have no influence on another variable (or set of variables) if our belief in the value of the latter does not change when learning something about the value of the former.

These two kinds of independence are not necessarily exclusive. Ideally, a good definition of independence expresses both the lack of causality (so it can be easily expressed by experts), and is useful for computations. In the probabilistic framework causal and decompositional independence relations are equivalent, which is not the case in possibility theory.

In this part, we first present non-classical theories of uncertainty and especially possibility theory (Chapter 1). Then we propose a qualitative uncertainty framework where uncertainty is represented by total pre-orders on the universe of discourse (Chapter 2). Finally we define the notion of qualitative independence and relate it to the basic existing independence relations relative to possibility theory (Chapter 3).

Chapter 1

Introduction to Possibility Theory

1.1 Introduction

The decision quality is closely related to the reliability of available information which is often imperfect due to the imprecision and uncertainty characterizing the real world. Probability theory is appropriate to model such information when all numerical data are available. However, it is not always possible to provide precise numerical values. Indeed, there are some situations which are not considered, in particular the case of *total ignorance* [61]. This is true in Bayesian networks when missing data does not allow any valid treatment.

Several non-classical theories of uncertainty have been proposed in order to deal with uncertain and imprecise data such as evidence theory [110, 117, 118], Lehmann's ranked model [96], plausibility relations [75], Spohn's ordinal conditional functions [119, 120] and possibility theory [60, 130] issued from fuzzy sets theory [86, 128].

Possibility theory, offers a natural and simple model to handle qualitative uncertain information. This theory has been developed from different aspects, we can mention [60],

- *Reasoning.* One of the tools used in reasoning is *possibilistic logic* which is an extension of classical logic where propositional or first-order formulas are weighted by lower bounds and necessity (or possibility) measures [52, 91].

- *Default reasoning.* Possibility theory also offers a general framework for dealing with rules having exceptions of the form "*generally, if p then q*" [19].

- *Qualitative decision.* Possibility theory offers a qualitative counterpart of classical decision. Namely, in [59], an analog of von Neumann and Morgenstern postulates, intended

for rational decision under ordinal uncertainty, has been proved to be equivalent to the maximization of a qualitative utility function.

- *Data Fusion.* Possibility theory offers a variety of combination modes (including weighted, prioritized and adaptative aggregation rules) in uncertain environments [20, 21, 57].

This Chapter proposes a brief overview of probability theory and presents basic definitions of non-classical theories of uncertainty. Section 1.2 introduces some notations and definitions used in this Chapter. Section 1.3 gives a brief recall on classical probability theory, evidence theory and Spohn's ordinal conditional functions. Lastly, Section 1.4 focuses on possibility theory and its main definitions and axioms.

1.2 Notations and definitions

We first give some notations and definitions. Let $V = \{A_1, A_2, ..., A_N\}$ be a set of variables,

- $D_A = \{a_1, ..., a_n\}$ denotes the supposedly finite domain associated with the variable A,
- a_i denotes any instance of A_i,
- $X, Y, Z, ...$, denote subsets of variables from V,
- $D_X = \times_{A_i \in X} D_{A_i}$ denotes the cartesian product of domains of variables in X,
- x denotes any instance of X, if $X = \{A_1, ..., A_n\}$ then $x = (a_1, ..., a_n)$,
- $\Omega = \times_{A_i \in V} D_{A_i}$ denotes the universe of discourse, which is the cartesian product of all variable domains in V,
- Each element $\omega \in \Omega$ is called an *interpretation*, a possible *world* or a *state* of Ω. Depending on the context, we use one of the following notations:
  - either tuples: $\omega = (a_1, ..., a_N)$
  - or conjunctions: $\omega = a_1 \wedge ... \wedge a_N$, then $\omega[A_i] = a_i$.
- ϕ, ψ, φ denote the subclasses of Ω (called propositions or events) and $\neg\phi$ denotes the complementary set of ϕ i.e. $\neg\phi = \Omega - \phi$,
- $\phi \wedge \psi$ (resp. $\phi \vee \psi$) denotes the intersection (resp. the union) of ϕ and ψ,
- $[a_i] = \{\omega = (a_1 \wedge ... \wedge a_N) : A_i = a_i\}$ denotes the set of states whose i^{th} component is a_i,

- $\forall x \in D_X, [x] = \{\omega = (a_1 \wedge ... \wedge a_n) : \forall A_i \in X = \{A_1, ..., A_n\}, A_i = a_i\}$ denotes the set of states whose restrictions to variables in X is x.

  When there is no ambiguity, we use x instead of $[x]$ and $x \wedge y$ (resp. $x \vee y$) instead of $[x] \wedge [y]$ (resp. $[x] \vee [y]$).

In the rest of this work, we will often refer to the following example to illustrate different notions:

Example 1.1 *Suppose that in a cultivated field, we have information about the physiological accidents that can affect the culture due to bacteria, mushrooms etc., the maintenance (chemical fertilizers, etc.) and the land yield, then:*

- *We can distinguish three variables i.e., physiological accidents (Pacc), maintenance (Maint) and land yield (Yield) thus* $V = \{PAcc, Maint, Yield\}$.

- *The domains associated with these variables are :*
  $D_{PAcc} = \{Disease1(d1), Disease2(d2), NoDisease(nd)\}$,
  $D_{Maint} = \{Good(gm), Medium(mm), Weak(wm)\}$,
  $D_{Yield} = \{Good(gy), Weak(wy)\}$.

  Note that, for the sake of simplicity, in some examples we only use binary variables. This will be made precise in each use.

- *The set of all states is* $\Omega = D_{PAcc} \times D_{Maint} \times D_{Yield} = \{d1 \wedge gm \wedge gy, d1 \wedge gm \wedge wy, d1 \wedge mm \wedge gy, d1 \wedge mm \wedge wy, d1 \wedge wm \wedge gy, d1 \wedge wm \wedge wy, d2 \wedge gm \wedge gy, d2 \wedge gm \wedge wy, d2 \wedge mm \wedge gy, d2 \wedge mm \wedge wy, d2 \wedge wm \wedge gy, d2 \wedge wm \wedge wy, nd \wedge gm \wedge gy, nd \wedge gm \wedge wy, nd \wedge mm \wedge gy, nd \wedge mm \wedge wy, nd \wedge wm \wedge gy, nd \wedge wm \wedge wy\}$.

- *A possible state is that there is no disease, and that the maintenance and the yield are good:* $\omega = nd \wedge gm \wedge gy$. *Then,* $\omega[PAcc] = nd, \omega[Maint] = gm$ *and* $\omega[Yield] = gy$.

- *The set* $[nd] = \{nd \wedge gm \wedge gy, nd \wedge gm \wedge wy, nd \wedge mm \wedge gy, nd \wedge mm \wedge wy, nd \wedge wm \wedge gy, nd \wedge wm \wedge wy\}$ *denotes the set of states where the instance nd of the variable PAcc holds.*
  The set $[gm] = \{d1 \wedge gm \wedge gy, d1 \wedge gm \wedge wy, d2 \wedge gm \wedge gy, d2 \wedge gm \wedge wy, nd \wedge gm \wedge gy, nd \wedge gm \wedge wy\}$ *denotes the set of states where the instance gm of the variable Maint holds.*
  The models of the event $nd \wedge gm$ *are* $[nd \wedge gm] = \{nd \wedge gm \wedge gy, nd \wedge gm \wedge wy\} = [nd] \cap [gm]$ *where* $\cap$ *is the set intersection symbol.*

1.3 Brief refresher on uncertainty frameworks

This Section, gives a brief recalling on the basic concepts of the classical theory on uncertainty and two of the non-classical theories, namely the evidence theory and Spohn's ordinal conditional functions.

1.3.1 Probability theory

The classical theory of probability is based on the notion of *probability distribution* which is a function $p : \Omega \rightarrow [0, 1]$ satisfying $\Sigma_{\omega \in \Omega} p(\omega) = 1$. Given a probability distribution p, we can define a *probability measure* of any subset $\phi \subseteq \Omega$ by $P(\phi) = \Sigma_{\omega \in \phi} p(\omega)$. Other concepts of probability theory are as follows:

- *Probabilistic conditioning*: in the probabilistic setting, a probability distribution p is transformed into a new probability distribution by the arrival of a new fully certain piece of information $\phi \subseteq \Omega$, as follows:

$$p(\omega \mid \phi) = \begin{cases} \frac{p(\omega)}{P(\phi)} & \text{if } \omega \in \phi \\ 0 & \textit{otherwise.} \end{cases} \tag{1.1}$$

- *Marginalization*: given a joint probability distribution p on Ω, we can derive the marginal distributions relative to subsets of variables using the **summation**. Then, $\forall X \subseteq V, \forall x \in D_X$:

$$P(x) = \sum_{\omega \in \Omega} \{p(\omega) : \omega[X] = x\}. \tag{1.2}$$

- *Bayes theorem*: this theorem provides a mathematical rule explaining how we should change our existing beliefs in the light of new information:

$$p(\omega \mid \phi) = \frac{P(\phi \mid \omega) \cdot p(\omega)}{P(\phi)}. \tag{1.3}$$

- *Conditional independence*: given three disjoint subsets of variables: X, Y and Z. The independence relation between the variable sets X and Y in the context Z, denoted by $I_{Prob}(X, Z, Y)$, is expressed by:

$$P(x \mid y \wedge z) = P(x \mid z), \forall x, y, z, \tag{1.4}$$

or equivalently,

$$P(x \wedge y \mid z) = P(x \mid z) \cdot P(y \mid z), \forall x, y, z. \tag{1.5}$$

This means that X is considered as Prob-independent from Y in the context Z if for any instance $z \in D_Z$, the probability degree of any $x \in D_X$ remains unchanged for any value $y \in D_Y$.

1.3.2 Evidence theory

The *evidence theory* [110, 117] encodes our knowledge by a *basic belief assignment* (b.b.a) which is a function $m : 2^\Omega \to [0,1]$ satisfying:
(1) $m(\emptyset) = 0$.
(2) $\Sigma_{\phi \subseteq \Omega}(\phi) = 1$.

The term $m(\phi)$, called the *basic belief mass* (b.b.m) assigned to ϕ, represents the part of a total and finite amount of belief that supports the fact that the actual world belongs to ϕ and does not support the fact that the actual world belongs to a strict subset of ϕ. If $m(\phi) > 0$, then ϕ is said to be a *focal element*. In the *Transferable Belief Model* framework [117, 118], condition 1 is not necessary required. If m has at most one focal element $\phi \neq \Omega$, $\phi \neq \emptyset$ i.e.
$m(\phi) = s, m(\Omega) = 1 - s, m(elsewhere) = 0, s \in [0,1]$,
then its related belief function is called a *simple support function*.

The total amount of belief committed to any event ϕ is expressed by a *belief function*: $Bel : 2^\Omega \to [0,1]$, defined for any $\phi \subseteq \Omega$ by:

$$Bel(\phi) = \Sigma_{\psi \subseteq \phi} m(\psi) = 1. \tag{1.6}$$

Related to Bel is the *plausibility function* $Pl : 2^\Omega \to [0,1]$, which quantifies the degree of plausibility that the actual world belongs to ϕ. For any $\phi \subseteq \Omega$ by $Pl(\phi)$ is expressed by:

$$Pl(\phi) = 1 - Bel(\neg\phi). \tag{1.7}$$

If all the focal elements of a basic belief assignment are singletons, then Bel is a probability measure, and $m = Bel = Pl$.

Several definitions of conditioning are developed in evidence theory, we give here the expression of the Dempster rule of conditioning:

$$Pl(\omega \mid \phi) = \frac{Pl(\omega)}{Pl(\phi)}. \tag{1.8}$$

1.3.3 Spohn's ordinal conditional functions

Ordinal conditional functions, known as *Kappa functions* [119, 120] encodes our knowledge on the real world (generally ill known) by a function $\kappa : \Omega \rightarrow \mathcal{N}$, where $\mathcal{N}$ is the set of natural numbers. Kappa functions are used in belief revision [126], default reasoning [72], and iterated belief revision [36].

A kappa distribution κ represents a grading of *disbelief* such that $\exists \omega \in \Omega$ s.t. $\kappa(\omega) = 0$. The more $\kappa(\omega)$ decreases, the more ω is preferred. Given any kappa distribution κ, a ranking can be defined on subsets ϕ of Ω by:

$$\kappa(\phi) = min\{\kappa(\omega) \mid \omega \in \phi\}. \tag{1.9}$$

Conditioning is defined by Spohn in the following way:

$$\kappa(\omega \mid \phi) = \begin{cases} \kappa(\omega) - \kappa(\phi) & if \omega \in \phi \\ +\infty & otherwise. \end{cases} \tag{1.10}$$

Moreover if $\phi \cap \psi \neq \emptyset$ then $\kappa(\psi \mid \phi) = min\{\kappa(\omega \mid \phi) \mid \omega \in \phi \wedge \psi\} = \kappa(\phi \wedge \psi) - \kappa(\phi)$.
From this definition it is easy to derive the conjunction axiom defined by:

$$\kappa(\phi \wedge \psi) = \kappa(\phi) + \kappa(\psi \mid \phi). \tag{1.11}$$

Kappa functions framework looks very similar to the probability theory [119, 120] since we only seem to have replaced the *minimum* by the *addition*, and the *addition* by the *multiplication*. Indeed, given an ordinary probability function P defined over Ω, $P(\omega)$ can be seen as a polynomial function of some small positive parameter z, for instance, α, $\beta \cdot z$, $\gamma \cdot z^2$, etc. Thus the probabilities assigned to any ϕ, and the conditional probabilities $P(\phi \mid \psi)$ will be rational functions of z. The function $\kappa(\phi \mid \psi)$ is defined as the lowest n such that $lim_{r \rightarrow 0} \frac{P(\phi|\psi)}{r^n} \neq 0$ which means that $\kappa(\phi \mid \psi) = n$ is of the same order of magnitude as $P(\phi \mid \psi)$.

1.4 Possibility theory

The possibility theory introduced by Zadeh [130] and developed by Dubois and Prade [60] handles uncertainty in a qualitative way, but encodes it in the interval $[0, 1]$ called possibilistic scale.

1.4.1 Possibility distribution

The basic building block in the possibility theory is the notion of *possibility distribution* denoted by π and corresponding to a mapping from Ω to the scale $[0, 1]$ encoding our knowledge

on the real world, denoted by u, which is generally ill known. The possibilistic scale can be interpreted in two manners:
- in an *ordinal* manner if the handled values reflect only an ordering between the different states of the world,
- in a *numerical* manner if the handled values make sense in the ranking scale.

Technically, a possibility distribution is a normal fuzzy set (at least one membership grade equal 1). For example all fuzzy numbers are possibility distributions. However, possibility theory can also be derived without reference to fuzzy sets. Table 1.1 presents some interpretations regarding to possibility distributions.

Table 1.1: Possibility distribution π

$\pi(\omega) = 0$	$\omega = u$ is impossible
$\pi(\omega) = 1$	$\omega = u$ is possible / unsurprising
$\pi(\omega) > \pi(\omega')$	$\omega = u$ is preferred to $\omega' = u$ (or is more plausible)

- *Normalization*: A possibility distribution π is said to be α-*normalized*, if its normalization degree, denoted $h(\pi)$, is equal to α, namely:

$$\alpha = h(\pi) = \max_{\omega} \pi(\omega). \tag{1.12}$$

  If $\alpha = 1$, then π is simply said to be *normalized*.

- *Marginalization*: Given a joint possibility distribution π on Ω, we can derive marginal distributions relative to subsets of variables using the **maximum** operator i.e. $\forall X \subseteq V, \forall x \in D_X$:

$$\pi(x) = max_{\omega \in \Omega}\{\pi(\omega) : \omega[X] = x\}. \tag{1.13}$$

- *Combination*: Given n joint possibility distributions π_1,...,π_n, on Ω_1,...,Ω_n, we can derive the joint possibility distribution on $\Omega_1 \times ... \times \Omega_n$ by combining them. There are different ways to combine possibility distributions (see [20, 57] for overview). In this work, we are interested in two forms of combination depending on the meaning of the possibilistic scale. Indeed, in an ordinal setting, we use the **minimum** operator to combine different distributions. However, in a numerical setting (if the definition makes sense in the ranking scale), we use the **product** operator to combine different distributions since this operator has no mean with values reflecting just an ordering between different states.

1.4.2 Possibility and necessity measures

In probability theory, the quantity $P(\neg\phi)$ is fully determined by $P(\phi)$ since $P(\neg\phi) = 1 - P(\phi)$. Hence, if ϕ *is not probable*, then $\neg\phi$ is *necessarily probable*.

However, the expression *"it is not possible that ϕ is true"* not only implies that *"$\neg\phi$ is possible"* but it also leads to a stronger conclusion i.e. *"it is necessary that $\neg\phi$"*. Moreover, the expression *"it is possible that ϕ is true"* does not entail anything about the possibility nor the impossibility of ϕ.

Thus, the description of uncertainty about the occurrence of ϕ needs two dual measures: the *possibility measure* $\Pi(\phi)$ and the *necessity measure* $N(\phi) = 1 - \Pi(\neg\phi)$ due to the weak relationship existing between these two quantities.

POSSIBILITY MEASURE: given a possibility distribution π, we can define a mapping grading the **possibility measure** of any subset $\phi \subseteq \Omega$ by:

$$\Pi(\phi) = max_{\omega \in \phi} \pi(\omega). \tag{1.14}$$

$\Pi(\phi)$ is called the possibility degree of ϕ, it corresponds to the possibility degree to have one of the models of ϕ as the real world. This measure evaluates at which level ϕ is **consistent** with our knowledge represented by π.

For instance, let us consider that we receive an information about a new breed, in the north pole, that the researches have named *Glacyria*. Thus for the question *what is the possibility that Glacyria has two legs ?* we can say that *it is fully possible* (i.e. its possibility degree is equal to 1) since we don't know *Glacyria* and we have no information that contradicts the fact that this animal has two legs. This response is consistent with our knowledge. Table 1.2 gives main properties of possibility measures.

NECESSITY MEASURE: the dual of the possibility measure of any subset $\phi \subseteq \Omega$ is the **necessity measure** defined by:

$$N(\phi) = 1 - \Pi(\neg\phi) = min_{\omega \notin \phi}(1 - \pi(\omega)). \tag{1.15}$$

$N(\phi)$ is called the necessity degree of ϕ, it corresponds to the certainty degree associated with ϕ i.e. the certainty degree to have one of the models of ϕ as the real world. In other terms, this measure evaluates at which level ϕ is **certainly** implied by our knowledge represented by π.

Table 1.2: Possibility measure Π (case of normalized possibility distributions)

$\Pi(\phi) = 1$ and $\Pi(\neg\phi) = 0$	ϕ is certainly true
$\Pi(\phi) = 1$ and $\Pi(\neg\phi) \in]0, 1[$	ϕ is somewhat certain
$\Pi(\phi) = 1$ and $\Pi(\neg\phi) = 1$	total ignorance (ϕ is unknown)
$\Pi(\phi) > \Pi(\psi)$	ϕ is a priori more plausible than ψ
$max(\Pi(\phi), \Pi(\neg\phi)) = 1$	ϕ and $\neg\phi$ cannot be both impossible (it is the unique link existing between $\Pi(\phi)$ and $\Pi(\neg\phi)$)
$\Pi(\phi \vee \psi) = max(\Pi(\phi), \Pi(\psi))$	decomposability axiom (disjunction axiom)
$\Pi(\phi \wedge \psi) \leq min(\Pi(\phi), \Pi(\psi))$	conjunction axiom

For instance suppose that I receive an e-mail from my colleague *Foulen* saying that he is in Paris. Then, if someone asks me *is Foulen here ?* I will say *no* since it is impossible that he is at the same time here and in Paris. This is equivalent to say that it is necessary (certain) that he is not here.

Table 1.3 gives main properties of necessity measures. Note that, $N(\phi) > 0 \Rightarrow \Pi(\phi) = 1$. This means that an event is completely possible before being somewhat certain. This property ensures the natural inequality $N(\phi) \leq \Pi(\phi)$.

Table 1.3: Necessity measure N (case of normalized possibility distributions)

$N(\phi) = 1$ and $N(\neg\phi) = 0$	ϕ is certain
$N(\phi) \in]0, 1[$ and $N(\neg\phi) = 0$	ϕ is somewhat certain
$N(\phi) = 0$ and $N(\neg\phi) = 0$	total ignorance (ϕ is unknown)
$min(N(\phi), N(\neg\phi)) = 0$	the unique relation existing between $N(\phi)$ and $N(\neg\phi)$
$N(\phi \wedge \psi) = min(N(\phi), N(\psi))$	conjunction axiom

1.4.3 Possibilistic conditioning

Conditioning is a crucial notion when studying independence relations. In the possibilistic setting it consists in modifying our initial knowledge, encoded by the possibility distribution π by the arrival of a new fully *certain* piece of information e. Let us denote $\phi = [e]$ the set of models of e. The initial distribution π is then replaced by another one denoted by $\pi' = \pi(. \mid \phi)$. Assuming that $\phi \neq \emptyset$ and that $\Pi(\phi) > 0$, the natural postulates for possibilistic conditioning are:

C_1: if $\pi(\omega) = 0$ then $\pi'(\omega) = 0$,

C_2: $\forall \omega \notin \phi, \pi'(\omega) = 0$,

C_3: π' should be normalized,

C_4: $\forall \omega_1, \omega_2 \in \phi, \pi(\omega_1) > \pi(\omega_2)$ iff $\pi'(\omega_1) > \pi'(\omega_2)$

C_5: if $\Pi(\phi) = 1$, then $\forall \omega \in \phi, \pi'(\omega) = \pi(\omega)$.

$\mathbf{C}_1$ says that irrelevant states remain irrelevant after conditioning, $\mathbf{C}_2$ confirms that ϕ is a fully certain piece of information and $\mathbf{C}_3$ says that the result should be a normalized possibility distribution. Moreover, $\mathbf{C}_4$ says that the new possibility distribution should not affect the order between the states in ϕ. Lastly, $\mathbf{C}_5$ says that if ϕ is already consistent with the beliefs encoded by π, then the possibility distribution remains unchanged on the models of ϕ. This is in agreement with the min-based combination mode which prevails in possibility theory; no further normalization is needed since $\Pi(\phi) = 1$.

Postulates ($\mathbf{C}_1$-$\mathbf{C}_5$) do not guarantee a *unique definition of conditioning*. Indeed, the effect of the axiom $\mathbf{C}_2$ may result in a sub-normalized possibility distribution, as shown by Example 1.2.

Note that, if we consider all the postulates except $\mathbf{C}_3$, then a possible definition of conditioning can be the following one:

$$\pi(\omega \mid_m \phi) = \begin{cases} \pi(\omega) & \text{if } \omega \in \phi \\ 0 & \textit{otherwise.} \end{cases} \tag{1.16}$$

Example 1.2 *Let us consider two binary variables, relative to climatic conditions (CCdt) and physiological accidents (PAcc), such that:*
$D_{CCdt} = \{Good(g), Bad(b)\}$
$D_{PAcc} = \{Yes(y), No(n)\}$ *with the joint possibility distribution*[1] *given in Table 5.6.*
Consider, now that we receive a fully certain piece of information indicating that there is a physiological accident ($[y] = \phi = \{b \wedge y, g \wedge y\}$). Then, using (1.16), the initial possibility distribution will be transformed into the one given in Table 1.5 (see Figure 1.1). Note that the resulting possibility distribution is sub-normalized.

Restoring the normalization, in order to satisfy $\mathbf{C}_3$, can be done in two different ways (when $\Pi(\phi) > 0$) depending on whether we are in a qualitative setting, where the scale $[0, 1]$

[1] *the instances in tables are ranked w.r.t. variables and their values. For instance, if we handle two binary variables A and B then the first (resp. second, third, fourth) instances corresponds to the first (resp. first, second, second) instance of the first variable with the first (resp. second, first, second) instance of the second one i.e.* $a_1b_1, a_1b_2, a_2b_1, a_2b_2$.

Table 1.4: Initial distribution

ccdt	*pacc*	$\pi(ccdt \wedge pacc)$
g	*y*	*0.4*
g	*n*	*1*
b	*y*	*0.8*
b	*n*	*0.8*

Table 1.5: New distribution

ccdt	*pacc*	$\pi(ccdt \wedge pacc \mid \phi)$
g	*y*	*0.4*
g	*n*	*0*
b	*y*	*0.8*
b	*n*	*0*

is only used for encoding an ordering between degrees (which may form a finite set of values), or if we are in a genuine numerical setting [54] (see Example 1.3):

- In an ordinal setting, we assign to the best elements of ϕ, the maximal possibility degree (i.e. 1), then we obtain:

$$\pi(\omega \mid_m \phi) = \begin{cases} 1 & if\ \pi(\omega) = \Pi(\phi) \text{ and } \omega \in \phi \\ \pi(\omega) & if\ \pi(\omega) < \Pi(\phi) \text{ and } \omega \in \phi \\ 0 & otherwise. \end{cases} \tag{1.17}$$

  This corresponds to the *min-based* conditioning.

- In a numerical setting (if the definition makes sense in the ranking scale), we proportionally shift up all elements of ϕ:

$$\pi(\omega \mid_p \phi) = \begin{cases} \frac{\pi(\omega)}{\Pi(\phi)} & \text{if } \omega \in \phi \\ 0 & otherwise. \end{cases} \tag{1.18}$$

  This corresponds to the *product-based* conditioning.

These two definitions of conditioning satisfy a unique equation close to the Bayesian rule, of the form:

$$\forall \omega,\ \pi(\omega) = \pi(\omega \mid \phi) \otimes \Pi(\phi). \tag{1.19}$$

Figure 1.1: Subnormalized possibilistic conditioning

respectively for $\otimes$ are the **minimum** (for (1.17)) and the **product** (for (1.18)) operators. The min-based conditioning (1.17) corresponds to the least specific solution of Equation (1.19) first proposed by Hisdal [78]. If $\Pi(\phi) = 0$ then, by convention $\pi(\omega \mid_m \phi) = \pi(\omega \mid_p \phi) = 1$.

Example 1.3 *Let us consider the sub-normalized possibility distribution $\pi(. \mid \phi)$ obtained in Example 1.2. We have, $[y] = \phi = \{b \wedge y, g \wedge y\}$, then $\Pi(\phi) = max(0.8, 0.4) = 0.8$.*

- *if we use the min-based conditioning expressed by (1.17), we obtain the possibility distribution given in Table 1.6.*

Table 1.6: Min-based conditioning

$ccdt$	$pacc$	$\pi(ccdt \wedge pacc \mid_m \phi)$
g	y	0.4
g	n	0
b	y	1
b	n	0

- *if we use the product-based conditioning expressed by (1.18), we obtain the distribution given in Table 1.7.*

Table 1.7: Product-based conditioning

$ccdt$	$pacc$	$\pi(ccdt \wedge pacc \mid_p \phi)$
g	y	0.5
g	n	0
b	y	1
b	n	0

The *normalization constraint* on conditional possibility distributions is defined as follows: Let X and Y be two subsets of V, then for any fixed instances x and y:

$$max\{\Pi(x \mid y), \Pi(\neg x \mid y)\} = 1. \tag{1.20}$$

1.4.4 Possibilistic Logic

Possibilistic logic [52, 91] handles qualitative uncertainty in a logical setting. A possibilistic logic formula is a pair (p, α) where p is a *propositional* or a *first-order* logic formula and α its uncertainty degree which estimates to what extent it is certain that p is true considering the available incomplete information. More formally, (p, α) encodes $N(p) \geq \alpha$. The higher is the

weight, the more certain is the formula. Possibilistic logic is essentially qualitative since only the pre-ordering on the formulas is important. Namely, $N(p) > N(q)$ means that the formula p is more certain than q.

A possibilistic knowledge base Σ is made up of a finite set of weighted formulas i.e.

$$\Sigma = \{(\phi_i, \alpha_i), i = 1, .., n\},$$

where α_i is the lower bound on the necessity degree $N(\phi_i)$. Formulas with $\alpha_i = 0$ are not explicitly represented in the knowledge base.

Given a possibilistic knowledge base Σ, a unique possibility distribution is generated by associating to each interpretation the level of compatibility with beliefs as follows:

$$\forall \omega \in \Omega, \pi_\Sigma(\omega) = \begin{cases} 1 & \text{if } \forall (\phi_i, \alpha_i) \in \Sigma, \omega \in [\phi_i] \\ 1 - max\{\alpha_i : (\phi_i, \alpha_i) \in \Sigma \text{ and } \omega \notin [\phi_i]\} & \text{otherwise.} \end{cases} \tag{1.21}$$

This means that all the interpretations satisfying all the beliefs in Σ will have the highest possibility degree, namely 1, and the other ones will be ranked w.r.t the most certain belief that they falsify.

Example 1.4 *Let r and s be two propositional symbols which stands, respectively, for* it rains, it snows. *Let $\Sigma = \{(r, 0.3), (r \vee s, 0.5)\}$. Then,*

$$\pi_\Sigma(r \wedge s) = \pi_\Sigma(r \wedge \neg s) = 1, \pi_\Sigma(\neg r \wedge s) = 0.7, \pi_\Sigma(\neg r \wedge \neg s) = 0.5.$$

The two states $r \wedge s$ and $r \wedge \neg s$ are the preferred ones since they are the only ones which are consistent with Σ. Moreover, $\neg r \wedge s$ is preferred to $\neg r \wedge \neg s$, since the highest belief falsified by $\neg r \wedge s$ (i.e. $(r, 0.3)$) is less certain than the highest belief falsified by $\neg r \wedge \neg s$ (i.e. $(r \vee s, 0.5)$).

The possibilistic inference in possibilistic logic is as efficient as in classical logic refutation by resolution and has been implemented in the form of an A*-like algorithm [52, 91].

1.4.5 Kappa functions vs Possibility theory

Clearly, kappa-functions and possibility theory are very closed. Indeed, rather than associate to each state ω a degree between [0, 1], we associate to it an integer $\kappa(\omega)$ in the set of natural numbers $\mathcal{N}$. Given a kappa ranking κ, the following equations show how to transform kappa-functions into possibility measures:

$$\pi_\kappa(\omega) = 2^{-\kappa(\omega)}, \Pi_\kappa(\phi) = 2^{-\kappa(\phi)}, \text{ and } N_\kappa(\phi) = 2^{-\kappa(\neg\phi)}. \tag{1.22}$$

These equations indicates that possibility transformations of kappa functions are valued on particular rational subset in [0,1]. We can easily check that Π_κ is a possibility measure. Indeed:

$$\Pi_\kappa(\phi \cup \psi) = 2^{-\kappa(\phi \cup \psi)} = 2^{-min(\kappa(\phi),\kappa(\psi))} = max(2^{-\kappa(\phi)}, 2^{-\kappa(\psi)}) = max(\Pi_\kappa(\phi), \Pi_\kappa(\psi)). \tag{1.23}$$

Moreover, the counterpart of Spohn's conditioning in possibility theory is:

$$\pi_{\kappa(.|\phi)}(\omega) = 2^{-\kappa(\omega|\phi)} = 2^{-\kappa(\omega)-\kappa(\phi)} = \frac{2^{-\kappa(\omega)}}{2^{-\kappa(\phi)}} = \frac{\pi_\kappa(\omega)}{\Pi_\kappa(\phi)}, \forall \omega \in \phi. \tag{1.24}$$

which corresponds to the product-based conditioning (see (1.18)). Clearly, kappa-functions can be recovered in possibility framework using the product-based conditioning. The converse transformation is only possible when $\kappa(\omega) = -log_2(\pi(\omega))$ takes its value in the set of natural numbers $\mathcal{N}$.

1.4.6 Evidence theory vs Possibility theory

In the evidence theory framework, if the focal elements $\phi_1, ..., \phi_n$ are nested (i.e., $\phi_1 \subseteq ... \subseteq \phi_n$), then the belief function Bel is called a *consonant belief function* and for all $\phi, \psi \subseteq \Omega$, we have:
$Bel(\phi \wedge \psi) = min(Bel(\phi), Bel(\psi))$; and
$Pl(\phi \vee \psi) = max(Pl(\phi), Pl(\psi))$

It is stated that in this case *belief functions* are *necessity measures* and *plausibility functions* are *possibility measures* i.e. $Bel = N$ and $Pl = \Pi$.

However, it is important to note that in this case, the Dempster rule of conditioning defined by (1.8) corresponds to the product-based conditioning defined by (1.18) and not to the min-based one defined by (1.17). This means that *Valuation Based Systems* [116] using belief functions can be used to encode possibilistic networks based on the product operator, but not those based on the minimum operator.

1.5 Conclusion

In this Chapter we have proposed a brief overview on probability theory and several of non-classical theories of uncertainty including *evidence theory*, *Spohn's ordinal conditional functions* and more particularly *possibility theory* which offers a natural model to handle qualitative uncertain information. In next Chapter, we propose a qualitative uncertainty framework

where uncertainty is represented by total pre-orders on possible states. Chapter 3 will detail the notion of independence in possibility theory and in its underlying qualitative framework.

Chapter 2

A New Qualitative Uncertainty Framework

2.1 Introduction

Generally, it is easier for experts to provide a preferential relation relative to each possible situation of the universe of discourse instead of exact numerical values as probabilities, possibilities, kappa functions etc. In this Chapter, we propose a new theoretical framework to model the uncertainty in a qualitative way. The basic idea of this representation is to equip the referential Ω with a *total pre-order*, instead of using the interval $[0, 1]$. This total pre-order corresponds to a *plausibility relation* on Ω and simply enables us to express that some situations are more plausible than others. Plausibility relations can be seen as qualitative counterpart of possibility distributions.

This Chapter also introduces a further definition which will be helpful in easily defining the notion of qualitative independence in Chapter 3, namely the notion of *accepted beliefs*. The proposed qualitative framework, also also known as scale-based framework, recovers basic definitions of classical possibility theory. We show, in particular, that the qualitative conditioning extends the notion of possibilistic conditioning.

Section 2.2 introduces the notion of ordinal uncertainty and its representation by plausibility relations. Then, Section 2.3 presents the qualitative conditioning. Section 2.4 defines the notion of *accepted beliefs* associated with plausibility relations. Lastly, Section 2.5 studies the major differences between the qualitative framework and the possibilistic one.
Proofs of this Chapter are given in Appendix A.

Principle results of this Chapter are published in [2, 5].

2.2 Ordinal uncertainty

Possibility theory, presented in the previous Chapter, can be described in a qualitative way. The basic idea of the qualitative representation of uncertainty is to equip the referential Ω with a **total pre-order**[1], also called a **weak order**, instead of using the scale $[0, 1]$. This total pre-order denoted $\geq_\pi$, corresponds to a **plausibility relation**, also called a **comparative possibility relation**, on Ω and simply enables us to express that some situations are more plausible than others. We denote $=_\pi$ (resp. $>_\pi$, $<_\pi$) the equality (resp. inequality) relation corresponding to $\geq_\pi$. Namely, the relation $\omega =_\pi \omega'$ (resp. $\omega >_\pi \omega'$, $\omega <_\pi \omega'$) means that ω is as plausible as (resp. more plausible than, less plausible than) ω'.

We now give some definitions regarding to plausibility relations:

- *Most plausible states*: Given $\varphi = \{\omega_1, .., \omega_n\} \subseteq \Omega$, the most plausible state(s) in the set φ is defined by $max(\varphi)$ s.t.

$$max(\varphi) = \{\omega_i : \omega_i \in \varphi, \nexists \omega_j \in \varphi \text{ s.t. } \omega_j >_\pi \omega_i\}. \tag{2.1}$$

- *Least plausible states*: Given $\varphi = \{\omega_1, .., \omega_n\} \subseteq \Omega$, the least plausible state(s) in the set φ is defined by $min(\varphi)$ s.t.

$$min(\varphi) = \{\omega_i : \omega_i \in \varphi, \nexists \omega_j \in \varphi \text{ s.t. } \omega_i >_\pi \omega_j\}. \tag{2.2}$$

- Given a relation $\geq_\pi$ on Ω, we can lift it to a *plausibility measure relation* defined on the subsets of Ω denoted $\geq_\Pi$ by (e.g., [43]):

$$\phi \geq_\Pi \psi \text{ iff } \forall \omega \in \psi, \exists \omega' \in \phi \text{ such that } \omega' \geq_\pi \omega. \tag{2.3}$$

  Namely, $\phi \geq_\Pi \psi$ holds if a best element in ϕ is preferred to best element(s) in ψ. In other terms:

$$\phi \geq_\Pi \psi \text{ iff } \exists \omega \in max(\phi), \omega' \in max(\psi) \text{ such that } \omega \geq_\pi \omega'.$$

  The idea behind the relation $\geq_\Pi$ is that the agent whose epistemic state is modeled by the plausibility relation $\geq_\pi$ evaluates events by their most plausible state considering that if ϕ occurs, then the expected situation is among the states in $max(\phi)$ because they are normal states.

[1] A relation $\geq$ on Ω is a total pre-order if $\geq$ is reflexive, transitive and for all ω_1, ω_2, we have either $\omega_1 \geq \omega_2$ or $\omega_2 \geq \omega_1$.

This qualitative representation of uncertainty is also used in several non-monotonic formalisms like Lehmann's ranked models [96], plausibility relations [75], possibility theory [60], Spohn's ordinal conditional functions [119, 120] and system of spheres of Lewis [74, 97]. In particular Spohn represents plausibility relations by means of well-ordered partitions $\{\phi_1, ..., \phi_p\}$ such that:

$$\forall i \in \{1, .., p\}, \forall \omega, \omega' \in \phi_i : \omega =_\pi \omega',$$

$$\forall i < j \text{ s.t. } i \in \{1, .., p\}, j \in \{1, .., p\}, \forall \omega \in \phi_i, \forall \omega' \in \phi_j : \omega >_\pi \omega',$$

that is $\phi_1 = max(\Omega), \phi_p = min(\Omega)$. Thus, ϕ_1 contains the most plausible states of the world. When $\phi_1 = \Omega$, the plausibility relation $\geq_\pi$ is *uniform* and expresses complete ignorance.

For any subset $X \subseteq V$, the projection of $\geq_\pi$ on D_X is denoted by $\geq_\pi^X$ and is defined by:

$$x \geq_\pi^X x' \text{ iff } [x] \geq_\Pi [x']. \tag{2.4}$$

If the projection of $\geq_\pi$ on D_X is uniform, then the agent is ignorant about the subset of variables X, or in other words, X is not informed, otherwise there is a proper subset $\phi_X^* \subseteq D_X$ of plausible values of X, such that $\phi_X^* = max(D_X)$.

The plausibility relations satisfies the characteristic property [43]:

$$\phi \geq_\Pi \psi \Rightarrow \phi \vee \varphi \geq_\Pi \psi \vee \varphi.$$

The dual necessity relation is defined by:

$$\phi \geq_N \psi \text{ iff } \neg\psi \geq_\Pi \neg\phi \text{ iff } max(\neg\psi) \geq_\Pi max(\neg\phi). \tag{2.5}$$

$\phi \geq_N \psi$ means that the agent is more certain about ϕ than about ψ.

2.3 Qualitative conditioning

In the qualitative setting, conditioning consists in focusing a plausibility relation $\geq_\pi$ on a subclass $\phi \subseteq \Omega$, on the basis of a new piece of fully certain information about a case at hand. A plausibility relation restricted to ϕ, denoted by $\geq_{\pi|\phi}$ and called *conditional plausibility relation*, is obtained for answering questions on the case at hand for which only ϕ is known. We denote $=_{\pi|\phi}$ (resp. $>_{\pi|\phi}$, $<_{\pi|\phi}$) the equality (resp. inequality) relation corresponding to $\geq_{\pi|\phi}$. Natural postulates for qualitative conditioning are:

$\mathbf{A}_1$: $\forall \omega_1, \omega_2 \in \phi, \omega_1 >_\pi \omega_2$ iff $\omega_1 >_{\pi|\phi} \omega_2$,

$\mathbf{A}_2$: $\forall \omega_1 \in \phi, \forall \omega_2 \notin \phi, \omega_1 >_{\pi|\phi} \omega_2$,

$\mathbf{A}_3$: $\forall \omega_1, \omega_2 \notin \phi, \omega_1 =_{\pi|\phi} \omega_2$.

$\mathbf{A}_1$ means that the new plausibility relation should not alter the initial order between the elements of ϕ. $\mathbf{A}_2$ confirms that each model of ϕ should be preferred to any model not belonging to ϕ. Finally, the last postulate $\mathbf{A}_3$ says that the elements not belonging to ϕ are irrelevant and should be in the same equivalence class.

Regarding to postulates $\mathbf{C}_1$-$\mathbf{C}_5$ characterizing possibilistic conditioning (see Section 1.4.3), $\mathbf{C}_1$ and $\mathbf{C}_3$ have no counterparts within $\mathbf{A}_1$-$\mathbf{A}_3$ due to the lack of the notions of *normalization* and *impossible states* in the qualitative setting. Postulate $\mathbf{C}_4$ corresponds to $\mathbf{A}_1$ and postulates $\mathbf{C}_2$ corresponds to $\mathbf{A}_2$ and $\mathbf{A}_3$. Moreover, contrary to $\mathbf{C}_1$-$\mathbf{C}_5$, the three postulates $\mathbf{A}_1$-$\mathbf{A}_3$ determine in a **unique** manner the new conditional plausibility relation $\geq_{\pi|\phi}$. Indeed, $\leq_{\pi|\phi}$ is obtained from $\leq_\pi$ by preserving the relative ordering between elements of ϕ, forcing elements which are outside ϕ to be equally plausible, but less plausible than any element of ϕ. More details on the differences existing between the possibilistic and qualitative frameworks are given in Section 2.5.

The construction of conditional plausibility relations $\geq_{\pi|\phi}$ is illustrated by the following example.

Example 2.1 *Let us consider two binary variables, relative to climatic conditions (CCdt) and physiological accidents (PAcc), such that* $D_{CCdt} = \{Good(g), Bad(b)\}$, $D_{PAcc} = \{Yes(y), No(n)\}$ *with the following plausibility relation:*

$$g \wedge n >_\pi b \wedge y =_\pi b \wedge n >_\pi g \wedge y.$$

Consider, now that we receive a fully certain piece of information indicating that there is an accident ($[y] = \{b \wedge y, g \wedge y\}$)*, then the initial plausibility relation will be modified into the following, unique, relation (see Figure 2.1):* $b \wedge y >_{\pi|\phi} g \wedge y >_{\pi|\phi} g \wedge n =_{\pi|\phi} b \wedge n$. *Indeed, from* $\mathbf{A}_2$*, we have* $b \wedge y >_{\pi|\phi} g \wedge n$, $b \wedge y >_{\pi|\phi} b \wedge n$, $g \wedge y >_{\pi|\phi} g \wedge n$ *and* $g \wedge y >_{\pi|\phi} b \wedge n$. *Moreover, from* $\mathbf{A}_3$*, we have* $b \wedge n =_{\pi|\phi} g \wedge n$. *Then, from* $\mathbf{A}_1$*, we have* $b \wedge y >_{\pi|\phi} g \wedge y$.

Note that we have a unique conditional plausibility relation contrary to the possibilistic case (see Example 1.3).

Figure 2.1: Qualitative conditioning

The conditional plausibility relation $\geq_{\pi|\phi}$ induces a conditional plausibility measure relation $\geq_{\Pi|\phi}$ between events simply defined as follows:

$$\alpha \geq_{\Pi|\phi} \beta \text{ iff } \alpha \wedge \phi \geq_{\Pi} \beta \wedge \phi.$$

This kind of conditioning completely ignores the previous ordering between elements outside ϕ. Viewed as a revision process, conditioning imposes that all states in $\neg\phi$ become impossible, because ϕ is learned to be absolutely true. This is different in what is usually used in belief revision [67]. Indeed, for instance natural belief revision [26, 119, 120], considers minimal change for taking ϕ into account. It simply consists in moving the best elements in ϕ to the top level, and leaving the order between other states unchanged.

In our example, Figure 2.2 illustrates natural belief revision. Indeed, $b \wedge y$ becomes the more plausible state in the new relation, since it is the best one in the models of y. Then, the second model $g \wedge y$ takes the second place and then the counter models i.e. $g \wedge n$ and $b \wedge n$ are less plausible than the models of y but save their initial, intrinsic, ordering. Another example of belief revision is Papini's approach [101] which is obtained from $\mathbf{A}_1$, $\mathbf{A}_2$ and the following postulate: [$\mathbf{A}_4$]: $\forall \omega_1, \omega_2 \notin \phi, \omega_1 >_{\pi} \omega_2$ iff $\omega_1 >_{\pi|\phi} \omega_2$.

In our example, this revision mode corresponds to Figure 2.3. Indeed, only the best model of y (i.e. $b \wedge y$) becomes the more plausible one in the new relation, while other states save their initial ordering.

Figure 2.2: Natural belief revision

Figure 2.3: Revision in Papini's approach

2.4 Accepted beliefs

We now introduce the notion of *accepted beliefs* which will be helpful for defining qualitative independence in Chapter 3. This notion has been proposed in the context of default reasoning in [58, 66]. We propose here a more detailed analysis.

2.4.1 Definitions

A proposition ϕ is said to be accepted by the agent with plausibility relation $\geq_\pi$, if and only if $\phi >_N \neg\phi$ [48]. In particular, the set $\{\phi \text{ s.t. } \phi >_N \neg\phi\}$ is deductively closed under the classical logic inference. In other words, the subclasses of Ω are splitted into three families: *accepted beliefs* ϕ such that $\phi >_\Pi \neg\phi$, *rejected beliefs* ϕ such that $\neg\phi >_\Pi \phi$ and *ignored beliefs* ϕ such that $\phi =_\Pi \neg\phi$. This trichotomy can be encoded as follows:

Definition 2.1 *The acceptance function associated with a plausibility relation $\geq_\pi$ denoted by $\mathbf{Acc}_{\geq_\pi}(.)$ assigns to each ϕ a value in $\{-1, 0, 1\}$ in the following way:*

$$\mathbf{Acc}_{\geq_\pi}(\phi) = \begin{cases} 1 & \text{if } \phi >_\Pi \neg\phi \\ -1 & \text{if } \neg\phi >_\Pi \phi \\ 0 & \text{if } \phi =_\Pi \neg\phi. \end{cases} \tag{2.6}$$

When $\mathbf{Acc}_{\geq_\pi}(\phi) = 1$ (resp. $\mathbf{Acc}_{\geq_\pi}(\phi) = -1$) we say that ϕ is accepted (resp. rejected). $\mathbf{Acc}_{\geq_\pi}(\phi) = \mathbf{Acc}_{\geq_\pi}(\neg\phi) = 0$, corresponds to the situation of total ignorance concerning ϕ, i.e., ϕ and $\neg\phi$ are equally plausible.

Lemma 2.1 *The acceptance function is equivalently defined as follows:*

$$Acc_{\geq_\pi}(\phi) = \begin{cases} 1 & \text{if } max(\Omega) \subseteq \phi \\ -1 & \text{if } max(\Omega) \subseteq \neg\phi \\ 0 & otherwise. \end{cases} \tag{2.7}$$

2.4.2 Properties of accepted beliefs

The following proposition summarizes the properties of the acceptance function $\mathbf{Acc}_{\geq_\pi}$:

Proposition 2.1 *The properties of the acceptance function $\mathbf{Acc}_{\geq_\pi}$ are:*

1. *It is* monotonic *i.e.* $\phi \subseteq \psi \Rightarrow \mathbf{Acc}_{\geq_\pi}(\phi) \leq \mathbf{Acc}_{\geq_\pi}(\psi)$.
2. $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = 1$ *iff* $\mathbf{Acc}_{\geq_\pi}(\phi) = 1$ *and* $\mathbf{Acc}_{\geq_\pi}(\psi) = 1$.
3. $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = min(\mathbf{Acc}_{\geq_\pi}(\phi), \mathbf{Acc}_{\geq_\pi}(\psi))$ *except if* $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = -1$ *and* $\mathbf{Acc}_{\geq_\pi}(\phi) = \mathbf{Acc}_{\geq_\pi}(\psi) = 0$.
4. $\mathbf{Acc}_{\geq_\pi}(\neg\phi) = -\mathbf{Acc}_{\geq_\pi}(\phi)$.
5. $\mathbf{Acc}_{\geq_\pi}(\phi \vee \psi) = max(\mathbf{Acc}_{\geq_\pi}(\phi), \mathbf{Acc}_{\geq_\pi}(\psi))$ *except if* $\mathbf{Acc}_{\geq_\pi}(\phi \vee \psi) = 1$ *and* $\mathbf{Acc}_{\geq_\pi}(\phi) = \mathbf{Acc}_{\geq_\pi}(\psi) = 0$.

Only the property 3 of this proposition is proved in the appendix A. Properties 1, 2 and 4 are obvious consequences of Lemma 2.1 and property 5 is trivial using properties 3 and 4.

Property 2 confirms that the logic of accepted unconditional events is *classical* logic since the acceptance of ϕ and the acceptance of ψ entails the acceptance of ϕ and ψ.

The function $\mathbf{Acc}_{\geq_\pi}$ can be extended in order to take into account a given context. Then, a *conditional acceptance function* denoted by $\mathbf{Acc}_{\geq_\pi}(.|.)$ is defined by:

$$\mathbf{Acc}_{\geq_\pi}(\phi \mid \psi) = \begin{cases} 1 & \text{if } \phi \wedge \psi >_\Pi \neg\phi \wedge \psi \\ -1 & \text{if } \neg\phi \wedge \psi >_\Pi \phi \wedge \psi \\ 0 & \text{if } \phi \wedge \psi =_\Pi \neg\phi \wedge \psi. \end{cases} \tag{2.8}$$

When $\mathbf{Acc}$ is defined on subsets of Ω, we talk about *plain beliefs*, while when it is defined on conditionals we talk about *conditional beliefs*. In a fixed context ψ, $\mathbf{Acc}_{\geq_\pi}(. \mid \psi)$ enjoys the same properties as function $\mathbf{Acc}_{\geq_\pi}$.

Example 2.2 *Let us consider two binary variables A and B with the following plausibility relation:*

$$a_1 \wedge b_1 >_\pi a_2 \wedge b_1 >_\pi a_1 \wedge b_2 =_\pi a_2 \wedge b_2,$$

then, for instance:
$\mathbf{Acc}_{\geq_\pi}(a_1) = 1, \mathbf{Acc}_{\geq_\pi}(a_2) = -1,$
$\mathbf{Acc}_{\geq_\pi}(b_1) = 1, \mathbf{Acc}_{\geq_\pi}(b_2) = -1,$
$\mathbf{Acc}_{\geq_\pi}(a_1 \mid b_1) = 1, \mathbf{Acc}_{\geq_\pi}(a_1 \mid b_2) = 0,$
$\mathbf{Acc}_{\geq_\pi}(a_2 \mid b_1) = -1, \mathbf{Acc}_{\geq_\pi}(a_2 \mid b_2) = 0.$

The plausibility relation $\geq_\pi$ determines $\mathbf{Acc}_{\geq_\pi}$ in a unique manner. The converse is not true. Namely, many plausibility relations can generate the same set of *plain beliefs*, i.e, we can have the same $\mathbf{Acc}_{\geq_\pi}$ on all events (including the states). Indeed, two plausibility relations induce the same plain beliefs if and only if they share the same set of most plausible states, as obviously stated by Lemma 2.1. The other parts of the relations may thus differ.

Counter-example 2.1 *Let us consider the following values of* $\mathbf{Acc}_{\geq_\pi}$ *relative to the two binary variables A and B:*
$\mathbf{Acc}_{\geq_\pi}(a_1) = \mathbf{Acc}_{\geq_\pi}(b_1) = 1,$
$\mathbf{Acc}_{\geq_\pi}(a_2) = \mathbf{Acc}_{\geq_\pi}(b_2) = -1,$
$\mathbf{Acc}_{\geq_\pi}(a_1 \vee b_1) = 1, \mathbf{Acc}_{\geq_\pi}(a_2 \vee b_1) = 1,$
$\mathbf{Acc}_{\geq_\pi}(a_1 \vee b_2) = 1, \mathbf{Acc}_{\geq_\pi}(a_2 \vee b_2) = -1,$
$\mathbf{Acc}_{\geq_\pi}(a_2 \wedge b_1) = -1, \mathbf{Acc}_{\geq_\pi}(a_2 \wedge b_2) = -1,$
$\mathbf{Acc}_{\geq_\pi}(a_1 \wedge b_2) = -1, \mathbf{Acc}_{\geq_\pi}(a_1 \wedge b_1) = 1.$

We can check that the two plausibility relations:
$a_1 \wedge b_1 >'_\pi a_2 \wedge b_1 >'_\pi a_1 \wedge b_2 ='_\pi a_2 \wedge b_2$, *and*
$a_1 \wedge b_1 >''_\pi a_2 \wedge b_1 >''_\pi a_1 \wedge b_2 >''_\pi a_2 \wedge b_2$,
generate the same information on the accepted beliefs than those given above i.e:

$$\mathbf{Acc}_{\geq_\pi} = \mathbf{Acc}_{\geq'_\pi} = \mathbf{Acc}_{\geq''_\pi}.$$

Note that the difference between $\geq'_\pi$ *and* $\geq''_\pi$ *concerns the ordering between the two elements* $a_1 \wedge b_2$ *and* $a_2 \wedge b_2$.

If we restrict the function $\mathbf{Acc}_{\geq_\pi}$ to Ω, we can distinguish three cases (we note $\mathbf{Acc}_{\geq_\pi}(\{\omega\}) = \mathbf{Acc}_{\geq_\pi}(\omega)$):

- $\mathbf{Acc}_{\geq_\pi}(\omega) = 1$: in this case, ω is the unique state such that $\omega >_\pi \omega', \forall \omega' \neq \omega \in \Omega$. The state ω is then called the *accepted state* since $\{\omega\} >_N \{\omega'\}$ as well for any $\omega' \neq \omega$. Note that, if $\exists \omega$ such that, $\mathbf{Acc}_{\geq_\pi}(\omega) = 1$, then $\forall \omega' \neq \omega, \mathbf{Acc}_{\geq_\pi}(\omega') = -1$.
- When $max(\Omega)$ contains more than one plausible instance then $\mathbf{Acc}_{\geq_\pi}(\omega) \leq 0, \forall \omega \in \Omega$. More precisely, $\forall \omega \in max(\Omega), \mathbf{Acc}_{\geq_\pi}(\omega) = 0$.
- $\mathbf{Acc}_{\geq_\pi}(\omega) = -1$ is equivalent to $\omega \notin max(\Omega)$, i.e. ω is not a plausible state.

So, the function $\mathbf{Acc}_{\geq_\pi}(\omega)$ on *states* only distinguish between the most plausible states (i.e. $\mathbf{Acc}_{\geq_\pi}(x) \geq 0$) and the less plausible ones ($\mathbf{Acc}_{\geq_\pi}(x) = -1$).

Interestingly, the restriction of $\mathbf{Acc}_{\geq_\pi}$ on Ω enables the function $\mathbf{Acc}_{\geq_\pi}$ to be reconstructed on all subsets of Ω. Indeed, $max(\Omega) = \{\omega \text{ s.t. } \mathbf{Acc}_{\geq_\pi}(\omega) = 1\} \cup \{\omega \text{ s.t. } \mathbf{Acc}_{\geq_\pi}(\omega) = 0\}$ (one of the sets is empty), and then it is enough to apply Lemma 2.1. So, $\mathbf{Acc}_{\geq_\pi}(\omega) = \mathbf{Acc}_{\geq'_\pi}(\omega), \forall \omega \in \Omega$, if and only if, $\mathbf{Acc}_{\geq_\pi}(\phi) = \mathbf{Acc}_{\geq'_\pi}(\phi), \forall \phi \subseteq \Omega$.

However, the set of all *conditional beliefs* determines in a *unique* manner a plausibility relation on Ω constructed in this way:

$$\omega_1 >_\pi \omega_2 \text{ iff } \mathbf{Acc}_{\geq_\pi}(\{\omega_1\} \mid \{\omega_1, \omega_2\}) = 1. \tag{2.9}$$

Example 2.3 *Let us consider the following conditional beliefs relative to the two binary variables A and B:*
$\mathbf{Acc}_{\geq_\pi}(a_1 \mid b_1) = 1, \mathbf{Acc}_{\geq_\pi}(a_1 \mid b_2) = 0, \mathbf{Acc}_{\geq_\pi}(a_2 \mid b_1) = -1,$
$\mathbf{Acc}_{\geq_\pi}(a_2 \mid b_2) = 0, \mathbf{Acc}_{\geq_\pi}(b_1 \mid a_1) = 1, \mathbf{Acc}_{\geq_\pi}(b_1 \mid a_2) = -1,$
$\mathbf{Acc}_{\geq_\pi}(b_2 \mid a_1) = -1, \mathbf{Acc}_{\geq_\pi}(b_2 \mid a_2) = 1.$

Using (2.8), these conditional beliefs induce:
$a_1 \wedge b_1 >_\pi a_2 \wedge b_1$, $a_1 \wedge b_2 =_\pi a_2 \wedge b_2$, $a_1 \wedge b_1 >_\pi a_2 \wedge b_1$, $a_2 \wedge b_2 =_\pi a_1 \wedge b_2$, $a_1 \wedge b_1 >_\pi a_1 \wedge b_2$, $a_2 \wedge b_2 >_\pi a_2 \wedge b_1$, $a_1 \wedge b_1 >_\pi a_1 \wedge b_2$, $a_2 \wedge b_2 >_\pi a_2 \wedge b_1$.

We can check that these relations induce the following, unique, plausibility relation i.e. :

$$a_1 \wedge b_1 >_\pi a_1 \wedge b_2 =_\pi a_2 \wedge b_2 >_\pi a_2 \wedge b_1.$$

Proposition 2.2 *The acceptance functions* $\mathbf{Acc}_{\geq_\pi}(.)$ *and the conditional acceptance function* $\mathbf{Acc}_{\geq_\pi}(.|.)$ *are related by the following Bayesian-like equation:*

$$\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = min(\mathbf{Acc}_{\geq_\pi}(\phi \mid \psi), \mathbf{Acc}_{\geq_\pi}(\psi)) \tag{2.10}$$

It may happen that $\mathbf{Acc}_{\geq_\pi}(\psi) = 1$ but $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = 0$ or 1 or -1.

In the following, we use $\mathbf{Acc}(.)$ (resp. $\mathbf{Acc}(.|.)$) instead of $\mathbf{Acc}_{\geq_\pi}(.)$ (resp. $\mathbf{Acc}_{\geq_\pi}(.|.)$) when there is no ambiguity.

2.5 Possibilistic framework vs Qualitative framework

Each possibility distribution π generates a unique plausibility relation $\geq_\pi$ defined by:

$$\omega \geq_\pi \omega' \text{ iff } \pi(\omega) \geq \pi(\omega'). \tag{2.11}$$

Example 2.4 *Let* π *be a possibility distribution defined by Table 2.1. The plausibility relation* $\geq_\pi$ *relative to* π *is:*

$$a_2 \wedge b_2 >_\pi a_1 \wedge b_1 =_\pi a_2 \wedge b_1 >_\pi a_1 \wedge b_2.$$

Table 2.1: Joint possibility distribution

a	b	$\pi(a \wedge b)$	$\pi'(a \wedge b)$
a_1	b_1	*0.9*	*0.7*
a_1	b_2	*0.2*	*0.5*
a_2	b_1	*0.9*	*0.7*
a_2	b_2	*1*	*1*

However, a plausibility relation corresponds to an infinity of possibility distributions as shown by the following example.

Example 2.5 *We can check that the possibility distribution π' given in Table 2.1 generates the same plausibility relation than π (in the same Table) (i.e. $\geq_\pi$).*

Note that if we define $\geq_\pi$ from π using (2.11) then:

$$\phi \geq_\Pi \psi \text{ iff } \Pi(\phi) \geq \Pi(\psi).$$

We now focus on the major differences between the possibility theory and the qualitative framework. There are at least three differences between using possibility distributions or plausibility relations:

- *Normalization*: in possibility theory, fully plausible states receive the grade 1 (i.e. $\exists \omega \in \Omega$ s.t. $\pi(\omega) = 1$) while there is no counterpart of this notion in the qualitative setting.
- *Existence of impossible states* graded to 0 in possibility theory, while all states are somewhat possible in the qualitative setting.
- *Commensurability* between uncertainty levels, where all rankings reflect grades in the same scale $[0, 1]$. As we will show in next Chapter, the commensurability property is crucial in the *decomposition* of some qualitative independence relations.

Note that these remarks are also true for other qualitative representation frameworks, like Spohn's ordinal conditional functions, and not only in the possibility theory.

The normalization and the existence of impossible states explain why there are several definitions of possibilistic conditioning while there is a unique definition in the qualitative setting. We now show that the qualitative conditioning extends the notion of possibilistic conditioning in the case of positive possibility distributions.

Qualitative conditioning vs Possibilistic conditioning. Let π be a positive possibility distribution (i.e. $\forall \omega, \pi(\omega) > 0$). Let $\geq_\pi$ be the plausibility relation derived from π using (2.11) ($\geq_\pi$ is unique). Let $\phi \subseteq \Omega$ be a new fully certain piece of information. Let us explain the link between qualitative conditioning and possibilistic conditioning.

It is clear that when applying the possibilistic conditioning, we will not affect the order between the models of ϕ (due to $\mathbf{C}_4$). In the same manner, the qualitative conditioning will not alter the initial order between the models of ϕ (due to $\mathbf{A}_1$). Moreover, the possibilistic conditioning, will decrease the possibility degree of the elements not belonging to ϕ to the degree 0 (due to $\mathbf{C}_2$). Hence, all models of ϕ are preferred to $\neg\phi$. In the same manner, the

qualitative conditioning will move the elements not belonging to ϕ in the same equivalence class which will be less preferred than the models of ϕ (due to $\mathbf{A}_2$ and $\mathbf{A}_3$).

The Possibilistic conditioning uses two additional postulates i.e. $\mathbf{C}_1$ and $\mathbf{C}_3$ which have no counterpart in qualitative conditioning. However their use will not affect the order of instances in the new conditional possibility distribution. Indeed, $\mathbf{C}_1$, have no effect since we consider only the case of positive possibility distributions. Moreover, $\mathbf{C}_3$, means that the new distribution should be normalized, thus if we use min-based conditioning (see (1.17)) we will just assign to the best models of ϕ the degree 1 which will not alter its position in the new possibility distribution. This is also true if we use product-based conditioning (see (1.18)) since we will just proportionally shift up the possibility degrees in in the new possibility distribution without modifying the order between instances.

Thus we deduce that when using possibilistic conditioning on a positive possibility distribution π (with the minimum operator or the product operator) the order of instances in the new conditional possibility distribution is the same than in the conditional plausibility relation computed from the plausibility relation induced from π (see Figure 2.4).

Figure 2.4: Qualitative conditioning vs possibilistic conditioning

Example 2.6 *Let us continue Example 1.3. We can check that the plausibility relations induced from the conditional possibility distributions obtained in Tables 1.6 and 1.7 (relative, respectively, to min-based conditioning and product-based conditioning) are the same than the one obtained using the qualitative conditioning (see Example 2.1).*

2.6 Conclusion

In this Chapter, we have proposed a qualitative uncertainty framework where uncertainty is represented by total pre-orders on possible states of the universe of discourse. We have also introduced the notion of *accepted beliefs* which will be helpful in easily defining the notion of qualitative independence in next Chapter. The proposed framework, recovers the classical (numerical) possibility theory. In particular, we have shown that the notion of qualitative conditioning extends the possibilistic conditioning.
Next Chapter studies different independence relations relative to the proposed qualitative

framework and compares them to the ones already existing in possibility theory [38, 39, 62, 63].

Chapter 3

Independence for Qualitative Uncertainty Framework

3.1 Introduction

The study of independence relations is central in multiple criteria analysis [89], in relational data decomposition [127], in uncertain reasoning based on Bayesian networks [84, 103] and in logical reasoning [15, 33, 90, 92]. There has been a considerable interest in Artificial Intelligence, in the last few years, for discussing independence in various representation frameworks, due to the success of Bayesian networks. Conditional independence relations between variables play an important role in the handling of uncertain information. From an operational point of view, two forms of independence can be distinguished:

- *decompositional independence* which ensures the decomposition of a joint distribution pertaining to tuples of variables into local distributions on smaller subsets of variables. The reasoning machinery can then work at a local level without losing any information.
- *causal independence* for expressing the lack of causality between variables. This form of independence is always characterized in semantic terms. Roughly speaking, a variable (or set of variables) is said to have no influence on another variable (or set of variables) if our belief in the value of the latter does not change when learning something about the value of the former.

Contrary to the decompositional independence, causal independence relations are not necessarily *symmetric*. In other words, if a variable A is independent of B, we are not sure that B is independent of A. These two kinds of independence relations are not necessarily mutually exclusive. Ideally, a good definition of independence expresses both the lack of causality (so

it can be easily expressed by experts), and is useful for computations.

In the probabilistic framework, two variables A and B are said to be decomposably independent if the joint probability on the range of (A, B) is the product of the probability distribution of A and the probability distribution of B, i.e., $P(A \wedge B) = P(A) \cdot P(B)$. Moreover, A and B are said to be causally independent if the probability of B given A is the same as the probability of B, i.e., $P(B \mid A) = P(B)$. In this framework causal and decompositional independence relations are equivalent.

In possibility theory, and more generally in total pre-orderings settings, the situation is different since causal and decompositional relations are not always equivalent. In this Chapter we investigate possible definitions of independence in two settings, using qualitative plausibility relations, or possibility distribution ranging on the scale $[0, 1]$.

Different works have been achieved on independence relations: de Campos and Huete [38, 39], Fonck [62, 63], Studený [121], de Cooman and Kerre [40], del Cerro and Herzig [41], Vejnarová [124]. However, results presented in this Chapter differ from the previous ones since the proposed independence relations are only based on the qualitative plausibility relations induced by possibility distributions.

This chapter is organized as follows: Section 3.2 proposes and investigates independence relations in qualitative framework where only the plausibility relations underlying the possibility distributions are used. Then, Section 3.3 presents the existing independence relations in possibility theory. Section 3.4 studies the effect of the commensurability in the decomposition of qualitative independence relations. Section 3.5 provides a comparative study between already known definitions of possibilistic independence and the ones proposed in this Chapter. Section 3.6 studies the graphoid properties of different independence relations. Lastly, Section 3.7 proposes a software allowing to test independence relations satisfied by any possibility distribution or plausibility relation.
Proofs of this Chapter are given in Appendix B.
Principle results of this Chapter are published in [2, 3, 4, 5, 6, 7].

3.2 Qualitative independence

In this Section we propose several causal and decomposable definitions of qualitative independence. We will see in Section 3.5 that contrary to the probabilistic case, the link between these two kinds of independence is not always obvious.

3.2.1 Causal qualitative independence

In the qualitative setting, independence relations can be thought of either in terms of *qualitative plausibility relations*, or in terms of *acceptance measures*. The two views can be related, as shown below where we present two possible definitions of causal independence. Basically, the variable set X is independent of Y if upon learning any instance of Y:
- the agent's beliefs on D_X i.e. the accepted (resp. rejected and ignored) instances of X are preserved, or
- the relative ordering between instances of X is preserved.

Belief-preserving independence

The first notion of causal independence in the ordinal setting is concerned with the preservation of accepted and rejected beliefs. A set of variables X can be considered as independent of Y in the context Z, if the accepted and rejected beliefs pertaining to X, held in the context Z, remain unchanged when some information about Y is obtained. Formally:

Definition 3.1 *Let $\geq_\pi$ be a plausibility relation defined on $\Omega = D_V$ and consider three mutually disjoint subsets of variables X, Y and Z forming a partition of V. X is said to be BP-independent (BP for Belief Preserving) of Y in the context Z, denoted by $I_{BP}(X, Z, Y)$, iff $\forall \phi_X \subseteq D_X, \forall \psi_Y \subseteq D_Y, \forall \varphi_Z \subseteq D_Z$,*

$$\mathbf{Acc}(\phi_X \mid \psi_Y \wedge \varphi_Z) = \mathbf{Acc}(\phi_X \mid \varphi_Z). \tag{3.1}$$

Compared with the notion of qualitative independence previously introduced [14, 45, 46], this definition is stronger in two extents: in [14, 45] only particular events are concerned; moreover the idea was (especially in reference [45]) to preserve accepted beliefs only and not rejected ones.

Note that contrary to the situation in probability theory, BP-independence is not symmetric as shown by the counter-example below.

Counter-example 3.1 Lack of symmetry property for I_{BP}
Let us consider two binary variables A and B with the following plausibility relation: $a_1 \wedge b_1 >_\pi a_1 \wedge b_2 >_\pi a_2 \wedge b_2 >_\pi a_2 \wedge b_1$.

Table 3.1, shows that $I_{BP}(A, \emptyset, B)$ *is true, namely* $\forall a, b, \mathbf{Acc}(a \mid b) = \mathbf{Acc}(a)$. *However,* $I_{BP}(B, \emptyset, A)$ *is false, for instance* $\mathbf{Acc}(b_1) = 1 \neq \mathbf{Acc}(b_1 \mid a_2) = -1$.

Table 3.1: Lack of symmetry property for I_{BP}

a	b	$Acc(a \mid b)$	$Acc(a)$	$Acc(b \mid a)$	$Acc(b)$
a_1	b_1	*1*	*1*	*1*	*1*
a_1	b_2	*1*	*1*	*-1*	*-1*
a_2	b_1	*-1*	*-1*	***-1***	***1***
a_2	b_2	*-1*	*-1*	*1*	*-1*

It is then clear that $I_{BP}(X, Z, Y)$ means that fixing any instance z of Z, the set $\{x$ s.t. $x \wedge y \wedge z$ *is a plausible instance in* $D_X \wedge y \wedge z\}$ does not depend on y. Hence, knowing some information about Y does not alter accepted beliefs about X in context Z.

Definition 3.1 is stated for all events defined by X, Y and Z, respectively, since **Acc** is not a decomposable function. Nevertheless, it is enough to state it with instances of X, Y and Z only as stated by the following proposition.

Proposition 3.1 *Let* $\geq_\pi$ *be a plausibility relation defined on* $\Omega = D_V$ *and consider three mutually disjoint subsets of variables* X, Y *and* Z *forming a partition of* V. *The relation* $I_{BP}(X, Z, Y)$ *is true, iff,* $\forall x, y, z$,

$$\mathbf{Acc}(x \mid y \wedge z) = \mathbf{Acc}(x \mid z). \tag{3.2}$$

We denote by I_{BPS} the symmetrized version[1] of BP-independence relation; i.e. the variable set X is said to be BPS-independent of Y in the context Z if:

$$(i)\mathbf{Acc}(x \mid y \wedge z) = \mathbf{Acc}(x \mid z) \text{ and}$$

$$(ii)\mathbf{Acc}(y \mid x \wedge z) = \mathbf{Acc}(y \mid z), \forall x, y, z. \tag{3.3}$$

The BPS-independence relation preserves the *plausible* instances of X given Y and Y given X in context Z, but does not preserve the relative *ordering* between instances of X (resp. Y) in the context Y (resp. X) (except when restricting to binary variables).

Example 3.1 *Let* A *and* B *be two BPS-independent variables with the following plausibility relation* $\geq_\pi$: $a_1 \wedge b_1 >_\pi a_2 \wedge b_1 >_\pi a_3 \wedge b_1 >_\pi a_1 \wedge b_2 >_\pi a_2 \wedge b_2 =_\pi a_3 \wedge b_2$.
By projection, the local plausibility relation relative to A *is then* $a_1 >_\Pi a_2 >_\Pi a_3$.

[1] In what follows the suffix S is used to denote the symmetrized version of non symmetric relations.

However, in the context b_2, we have $a_1 >_\Pi a_2 =_\Pi a_3$, thus, the relative ordering between instances of A is not preserved in all contexts of B since $a_2 >_\Pi a_3$ while $a_2 =_\Pi a_3$ in the context b_2.

Order-preserving independence

The causality-oriented definition that we propose now simply says that X is independent of Y in the context of Z, if for all $z \in D_Z$, the local *preferential ordering* between the different instances of X is preserved after the revision by any instance y of Y. More formally:

Definition 3.2 *Let $\geq_\pi$ be a plausibility relation defined on three disjoint subsets of variables: X, Y and Z. The variable set X is said to be PO-independent (PO for Preserving Ordering) of Y in the context Z, denoted $I_{PO}(X, Z, Y)$, if $\forall z \in D_Z, \forall y \in D_Y$:*

$$\forall x_i, x_j \in D_X, x_i \wedge z >_\Pi x_j \wedge z \text{ iff } x_i \wedge y \wedge z >_\pi x_j \wedge y \wedge z. \tag{3.4}$$

Proposition 3.2 *If X is PO-independent of Y in the context Z, then X is also BP-independent of Y in the same context. The converse in not true.*

Counter-example 3.2 *Let us consider a ternary variable A and a binary variable B with the following plausibility relation: $a_1 \wedge b_1 >_\pi a_2 \wedge b_1 >_\pi a_3 \wedge b_1 >_\pi a_1 \wedge b_2 >_\pi a_2 \wedge b_2 =_\pi a_3 \wedge b_2$. We can check that A is BP-independent of B, but not PO-independent of B since the local plausibility relation relative to A is $a_1 >_\Pi a_2 >_\Pi a_3$. However, in the context b_2, we have $a_2 =_\Pi a_3$, thus the relation $I_{PO}(A, \emptyset, B)$ is false, since the ordering between a_2 and a_3 is not preserved in context b_2.*

Note that this relation is not symmetric as shown by the following counter-example:

Counter-example 3.3 *Let us consider two binary variables A and B with the following plausibility relation: $a_1 \wedge b_1 >_\pi a_1 \wedge b_2 >_\pi a_2 \wedge b_2 >_\pi a_2 \wedge b_1$.*

- *The local plausibility relation relative to A is $a_1 >_\Pi a_2$. Moreover, in the context b_1 (resp. b_2), we have $a_1 >_\Pi a_2$ since $a_1 \wedge b_1 >_\Pi a_2 \wedge b_1$ (resp. $a_1 \wedge b_2 >_\Pi a_2 \wedge b_2$). Thus, the relation $I_{PO}(A, \emptyset, B)$ is true since the ordering relative to the different instances of A is preserved for all instances of B.*

- *The local plausibility relation relative to B is $b_1 >_\Pi b_2$. However, in the context a_2, we have $b_2 >_\Pi b_1$, thus, the relation $I_{PO}(B, \emptyset, A)$ is false, since the ordering between b_1 and b_2 is not preserved in the context a_2.*

We denote I_{POS} the symmetrized version of I_{PO}; i.e. X is said to be POS-independent of Y in the context Z if $\forall z \in D_Z, \forall y \in D_Y, \forall x \in D_X$:

$$\text{(i) } \forall x_i, x_j \in D_X, x_i \wedge z >_\Pi x_j \wedge z \text{ iff } x_i \wedge y \wedge z >_\pi x_j \wedge y \wedge z, \text{ and}$$

$$\text{(ii) } \forall y_k, y_l \in D_Y, y_k \wedge z >_\Pi y_l \wedge z \text{ iff } x \wedge y_k \wedge z >_\pi x \wedge y_l \wedge z. \tag{3.5}$$

The following proposition rewrites POS-independence in terms of **Acc**.

Proposition 3.3 *X is POS-independent of Y in the context Z iff:*
$\forall D'_X \subseteq D_X, \forall D'_Y \subseteq D_Y$ such that $D'_X \neq \emptyset$ and $D'_Y \neq \emptyset$ and $\forall x, y, z$

$$\mathbf{Acc}(x \wedge y \mid z, D'_X, D'_Y) = min(\mathbf{Acc}(x \mid z, D'_X), \mathbf{Acc}(y \mid z, D'_Y)). \tag{3.6}$$

From this rewriting, we deduce the following proposition:

Proposition 3.4 *If X is POS-independent of Y in the context Z, then X is also BPS-independent of Y in the same context. The converse in not true.*

Counter-example 3.4 *: I_{BPS}* DOES NOT IMPLY *I_{POS}*

Let us consider a ternary variable A and a binary variable B with the following plausibility relation: $a_1 \wedge b_1 >_\pi a_2 \wedge b_1 >_\pi a_3 \wedge b_1 >_\pi a_1 \wedge b_2 >_\pi a_2 \wedge b_2 =_\pi a_3 \wedge b_2$.
We can check that A is BPS-independent of B, but not POS-independent of B since the local plausibility relation relative to A is $a_1 >_\Pi a_2 >_\Pi a_3$. However, in the context b_2, we have $a_2 =_\Pi a_3$, thus, the relation $I_{POS}(A, \emptyset, B)$ is false, since the ordering between a_2 and a_3 is not preserved in the context b_2.

We now compare the POS-independence relation to the well known independence relation based on *Ceteris Paribus* (all else being equal) principle used in [27, 42] and defined by:

Definition 3.3 *Let $\geq_\pi$ be a plausibility relation defined on three disjoint subsets of variables: X, Y and Z. The variable set X is said to be CP-independent (CP for Ceteris Paribus) of Y in the context Z, denoted $I_{CP}(X, Z, Y)$, if $\forall z \in D_Z, \forall x_i, x_j \in D_X, \forall y_k, y_l \in D_Y$,*

$$x_i \wedge y_k \wedge z >_\pi x_j \wedge y_k \wedge z \text{ iff } x_i \wedge y_l \wedge z >_\pi x_j \wedge y_l \wedge z. \tag{3.7}$$

Proposition 3.5 *CP-independence relation is equivalent to POS-independence relation.*

3.2.2 Decompositional independence

This section proposes two classes of decompositional independencies, the first is based on belief decomposition and the second on remarkable plausibility relations.

Belief decompositional independence

The idea of this independence relation is to consider two variable sets X and Y as independent in the context Z if for any instance z of Z, the acceptance of any instance $(x \wedge y)$ of (X, Y) is fully determined by the acceptance of x and y. One way to relate the acceptance of $(x \wedge y)$ to the acceptance of x and the acceptance of y is:

Definition 3.4 *Let $\geq_\pi$ be a plausibility relation defined on $\Omega = D_V$ and consider three mutually disjoint subsets of variables X, Y and Z forming a partition of V. X and Y are said to be PT-independent (PT for Preserving Top elements) in the context Z, denoted by $I_{PT}(X, Z, Y)$, iff $\forall \phi_X \subseteq D_X, \forall \psi_Y \subseteq D_Y, \forall \xi_Z \subseteq D_Z$*

$$\mathbf{Acc}(\phi_X \wedge \psi_Y \mid \xi_Z) = min(\mathbf{Acc}(\phi_X \mid \xi_Z), \mathbf{Acc}(\psi_Y \mid \xi_Z)). \tag{3.8}$$

This definition is analogous to the one given in probability theory i.e. two variables A and B are independent if the probability over A and B is fully determined by $P(A)$ and $P(B)$ (i.e. $P(A \wedge B) = P(A) \cdot P(B)$).

Proposition 3.6 *X and Y are PT-independent in the context Z as soon as Definition 3.4 holds for all instances of X, Y and Z only, that is:*

$$\forall x, y, z, \mathbf{Acc}(x \wedge y \mid z) = min(\mathbf{Acc}(x \mid z), \mathbf{Acc}(y \mid z)). \tag{3.9}$$

It means that the set of plausible instances of a cartesian product of domains is a cartesian product. In particular, if any of the two sets $max(D_X)$ and $max(D_Y)$ contains a simple element then, obviously, X and Y are PT-independent. So PT-independent is a very weak definition of independence (see Figure 3.1 in Section 3.5). In other terms, the acceptance of one instance of X or of Y is enough to conclude the independence between these two variable sets in the context Z:

Proposition 3.7 *$\forall z \in D_Z$, if $\exists x \in D_X$ such that $\mathbf{Acc}(x \mid z) = 1$ or $\exists y \in D_Y$ such that $\mathbf{Acc}(y \mid z) = 1$, then the relation $I_{PT}(X, Z, Y)$ is true.*

In particular, if a plausibility relation $\geq_\pi$ contains exactly one preferred element then all variables are pairwise PT-independent. Moreover, we have the following strong result:

Proposition 3.8 *Let X, Y, Z be three disjoint subsets of variables, then $\forall x, y, z$,*

$$\mathbf{Acc}(x \wedge y \mid z) \neq min(\mathbf{Acc}(x \mid z), \mathbf{Acc}(y \mid z))$$

$$\Leftrightarrow \mathbf{Acc}(x \wedge y \mid z) = -1, \mathbf{Acc}(x \mid z) = 0, \text{ and } \mathbf{Acc}(y \mid z) = 0.$$

The preservation of accepted beliefs, implicitly, implies the preservation of preferred instances but the converse is not true as stated by the following proposition.

Proposition 3.9 *If X is BPS-independent of Y in the context Z, then X and Y are also PT-independent. The converse in not true.*

Counter-example 3.5 *: I_{PT}* DOES NOT IMPLY *I_{BPS}*

Let us consider two binary variables A and B with the following plausibility relation:

$$a_1 \wedge b_2 >_\pi a_2 \wedge b_2 >_\pi a_1 \wedge b_1 =_\pi a_2 \wedge b_1,$$

we can check that A and B are PT-independent, but not BPS-independent.

Using Propositions 3.9 and 3.4, we deduce that if X is POS-independent of Y in the context Z, then X and Y are also PT-independent and that the converse is not true.

Decompositional independence of remarkable plausibility relations

A natural way of defining decompositional independencies is to analyze the structure of the plausibility relation $\geq_\pi$. Indeed, a plausibility relation is said to be decomposable w.r.t. X and Y in the context Z, iff $\geq_\pi$ is a function of the local orderings on $(X \cup Z)$ and $(Y \cup Z)$. The following introduces a well known principle, called *Pareto-principle*:

Definition 3.5 *Let $\geq_\pi$ be a plausibility relation and u_i, v_i be two instances (not necessarily different) of A_i. Let $\overrightarrow{u} = (u_1, ..., u_n)$ and $\overrightarrow{v} = (v_1, ..., v_n)$ be two vectors. Then, $\overrightarrow{u}$ is said to be weakly Pareto-preferred to $\overrightarrow{v}$, denoted by $\overrightarrow{u} \geq_P \overrightarrow{v}$, if and only if: $\forall u_i, \forall v_i, i \in \{1, .., n\}, u_i \geq_\Pi v_i$. Moreover, $\overrightarrow{u}$ is said to be strictly Pareto-preferred to $\overrightarrow{v}$, if and only if: $\overrightarrow{u} \geq_P \overrightarrow{v}$ and $\exists i \in \{1, .., n\}$ s.t. $u_i >_\Pi v_i$.*

In general $\geq_P$ is only a **partial** order. Since we deal with plausibility relations which are complete pre-orders, the following definition introduces a general class of plausibility relations which are compatible with the Pareto-principle:

Definition 3.6 *Let X, Y and Z be disjoint subsets of variables. A plausibility relation $\geq_\pi$ is said to be Pareto-compatible (or* **monotonic***) on X and Y in the context Z if $\forall z \in D_Z, \forall x_i, x_j \in D_X, \forall y_k, y_l \in D_Y$, we have:*
$(x_i \wedge z, y_k \wedge z) \geq_P (x_j \wedge z, y_l \wedge z)$ implies $(x_i \wedge y_k \wedge z) \geq_\pi (x_j \wedge y_l \wedge z)$

Well known example of orderings used in the qualitative setting, which are Pareto-compatible are the *Pareto*, the *leximin* and the *leximax* orderings that we briefly present now [98].

Definition 3.7 *Let* $\overrightarrow{u} = \{u_1, ..., u_n\}$ *and* $\overrightarrow{v} = \{v_1, ..., v_n\}$ *be two vectors, and let* σ *and* τ *be two permutations of indices such that* $\forall i \in \{1, .., n\}, u_{\sigma(i)} >_\Pi u_{\sigma(i+1)}$ *and* $v_{\tau(i)} >_\Pi v_{\tau(i+1)}$. *Then,*
- $\overrightarrow{u}$ *is said to be leximin-preferred to* $\overrightarrow{v}$, *denoted by* $\overrightarrow{u} >_{leximin} \overrightarrow{v}$, *if and only if there exists* i *such that* $u_{\sigma(i)} >_\Pi v_{\tau(i)}$ *and* $\forall j > i, u_{\sigma(j)} =_\Pi v_{\tau(j)}$.
- $\overrightarrow{u}$ *is said to be leximin-equal to* $\overrightarrow{v}$, *denoted by* $\overrightarrow{u} =_{leximin} \overrightarrow{v}$, *if and only if* $\forall i$, $u_{\sigma(i)} =_\Pi v_{\tau(i)}$.

The leximin ordering is a natural extension of the minimum operator which has been used in different areas like in handling conflicts in knowledge bases [13, 95], and in flexible constraint satisfaction problems [47, 49, 50].

Definition 3.8 *Let* $\overrightarrow{u} = \{u_1, ..., u_n\}$ *and* $\overrightarrow{v} = \{v_1, ..., v_n\}$ *be two vectors, and let* σ *and* τ *be two permutations of indices such that* $\forall i \in \{1, .., n\}, u_{\sigma(i)} >_\Pi u_{\sigma(i+1)}$ *and* $v_{\tau(i)} >_\Pi v_{\tau(i+1)}$. *Then,*
- $\overrightarrow{u}$ *is said to be leximax-preferred to* $\overrightarrow{v}$, *denoted by* $\overrightarrow{u} >_{leximax} \overrightarrow{v}$, *if and only if there exists* i *such that* $u_{\sigma(i)} >_\Pi v_{\tau(i)}$ *and* $\forall j < i, u_{\sigma(j)} =_\Pi v_{\tau(j)}$.
- $\overrightarrow{u}$ *is said to be leximax-equal to* $\overrightarrow{v}$, *denoted by* $\overrightarrow{u} =_{leximin} \overrightarrow{v}$, *if and only if* $\forall i$, $u_{\sigma(j)} =_\Pi v_{\tau(i)}$.

We now use these orderings to characterize plausibility relations:

1. A plausibility relation $\geq_\pi$ is said to be **Pareto-decomposable** on X and Y in the context Z, if $\forall z \in D_Z, \forall x_i, x_j \in D_X, \forall y_k, y_l \in D_Y$, we have:
   $x_i \wedge y_k \wedge z \geq_\pi x_j \wedge y_l \wedge z$ **if and only if**
   $x_i \wedge z \geq_\Pi x_j \wedge z$ and $y_k \wedge z \geq_\Pi y_l \wedge z$.

   This definition is very strong, in the sense that $\geq_\pi$ is Pareto-decomposable along X and Y if one of the groups of variables is not informed as stated by the following proposition:

   Proposition 3.10 *A plausibility relation* $\geq_\pi$ *is Pareto-decomposable on* X *and* Y *if one of the local plausibility relations on* X *or* Y *is* uniform. *The converse is not true.*

   Counter-example 3.6 *Let us consider the following plausibility relation pertaining to two binary variables* A *and* B:

   $$a_1 \wedge b_1 =_\pi a_2 \wedge b_1 >_\pi a_1 \wedge b_2 >_\pi a_2 \wedge b_2$$

   We can check that the local plausibility relations on A *is uniform (i.e.* $a_1 =_\Pi a_2$*) while the plausibility relation on* A *and* B *is not Pareto-decomposable.*

2. A plausibility relation $\geq_\pi$ is said to be **leximin-decomposable** on X and Y in the context Z, if $\forall z \in D_Z, \forall x_i, x_j \in D_X, \forall y_k, y_l \in D_Y$, we have:
   - $x_i \wedge y_k \wedge z >_\pi x_j \wedge y_l \wedge z$ **if and only if**
   (i) $min(x_i \wedge z, y_k \wedge z) >_\Pi min(x_j \wedge z, y_l \wedge z)$ or
   (ii) $min(x_i \wedge z, y_k \wedge z) =_\Pi min(x_j \wedge z, y_l \wedge z)$ and $max(x_i \wedge z, y_k \wedge z) >_\Pi max(x_j \wedge z, y_l \wedge z)$.
   - $x_i \wedge y_k \wedge z =_\pi x_j \wedge y_l \wedge z$ **if and only if**
   $min(x_i \wedge z, y_k \wedge z) =_\Pi min(x_j \wedge z, y_l \wedge z)$ and $max(x_i \wedge z, y_k \wedge z) =_\Pi max(x_j \wedge z, y_l \wedge z)$.

3. A plausibility relation $\geq_\pi$ is said to be **leximax-decomposable** on X and Y in the context Z, if $\forall z \in D_Z, \forall x_i, x_j \in D_X, \forall y_k, y_l \in D_Y$, we have:
   - $x_i \wedge y_k \wedge z >_\pi x_j \wedge y_l \wedge z$ **if and only if**
   (i) $max(x_i \wedge z, y_k \wedge z) >_\Pi max(x_j \wedge z, y_l \wedge z)$ or
   (ii) $max(x_i \wedge z, y_k \wedge z) =_\Pi max(x_j \wedge z, y_l \wedge z)$ and $min(x_i \wedge z, y_k \wedge z) >_\Pi min(x_j \wedge z, y_l \wedge z)$.
   - $x_i \wedge y_k \wedge z =_\pi x_j \wedge y_l \wedge z$ **if and only if**
   $min(x_i \wedge z, y_k \wedge z) =_\Pi min(x_j \wedge z, y_l \wedge z)$ and $max(x_i \wedge z, y_k \wedge z) =_\Pi max(x_j \wedge z, y_l \wedge z)$.

where $max(a,b) = \begin{cases} a \text{ if } a \geq_\Pi b \\ b \text{ otherwise} \end{cases}$

Definition 3.9 *X and Y are said to be **Pareto-independent** (resp. **leximin-independent**, **leximax-independent**) in the context Z, denoted I_{Pareto} (resp. $I_{leximin}$, $I_{leximax}$), if the plausibility relation $\geq_\pi$ is Pareto-decomposable (resp. leximin-decomposable, leximax-decomposable) on X and Y in the context Z.*

Proposition 3.11 *If X and Y are Pareto-independent in the context Z, then they are leximin-independent and leximax-independent . The converse is false and leximax independence is not comparable with leximin independence.*

Counter-example 3.7 *: $I_{leximin}$* AND *$I_{leximax}$* DO NOT IMPLY *I_{Pareto}* AND THEY ARE INCOMPRABLE

Let us consider the following plausibility relations pertaining to a binary variable A and ternary variable B:

$$a_1 \wedge b_1 >_\pi a_1 \wedge b_2 >_\pi a_2 \wedge b_1 >_\pi a_1 \wedge b_3 >_\pi a_2 \wedge b_2 >_\pi a_2 \wedge b_3,$$

$$a_1 \wedge b_1 >'_\pi a_1 \wedge b_2 >'_\pi a_2 \wedge b_1 >'_\pi a_2 \wedge b_2 >'_\pi a_1 \wedge b_3 >'_\pi a_2 \wedge b_3,$$

with $\geq_\pi$ we can check that A and B are leximax-independent but neither leximin-independent, since $a_1 \wedge b_3 >_\pi a_2 \wedge b_2$ while $min(a_2, b_2) >_\Pi min(a_1, b_3)$, nor Pareto-independent, since

$a_1 \wedge b_3 >_\pi a_2 \wedge b_2$ *while* $b_2 >_\Pi b_3$.

In addition with $\geq'_\pi$ *we can check that A and B are leximin-independent but neither leximax-independent since* $a_2 \wedge b_2 >'_\pi a_1 \wedge b_3$ *while* $max(a_1, b_3) >'_\Pi max(a_2, b_2)$, *nor Pareto-independent since* $a_2 \wedge b_2 >'_\pi a_1 \wedge b_3$ *while* $a_1 >'_\Pi a_2$.

Proposition 3.12 *Pareto, leximin and leximax independence imply POS-independence. The converse is false.*

Counter-example 3.8 *:* I_{POS} DOES NOT IMPLY $I_{leximin}$, $I_{leximax}$ AND I_{Pareto}

Let A and B be two variables and $\geq_\pi$, $\geq'_\pi$ *be the plausibility relations given in Counter example 3.7.*

- *with* $\geq_\pi$, *we can check that A is POS-independent of B but that these two variables are not leximin-independent since* $a_1 \wedge b_3 >_\pi a_2 \wedge b_2$ *while* $min(a_2, b_2) >_\Pi min(a_1, b_3)$. *Moreover, with* $\geq'_\pi$ *we can check that A is POS-independent of B but these two variables are not leximax-independent since* $a_2 \wedge b_2 >'_\pi a_1 \wedge b_3$ *while* $max(a_1, b_3) >'_\Pi max(a_2, b_2)$.

- *with the following plausibility relation:*

$$a_1 \wedge b_1 >_\pi a_1 \wedge b_2 >_\pi a_2 \wedge b_1 >_\pi a_2 \wedge b_2,$$

we can check that the relation $I_{POS}(A, \emptyset, B)$ *is true contrary to* $I_{Pareto}(A, \emptyset, B)$.

However, there are particular cases where the independence relations POS, leximin and leximax are equivalent:

- The first one concerns binary variables:

  Proposition 3.13 *If A and B are binary variables then A is POS-independent of B in the context of a binary variable C if and only if they are leximin-independent and if and only if they are leximax-independent.*

- The second one concerns two-level distributions:

  Proposition 3.14 *If* $\geq_\pi$ *is a two-level distribution, then X is POS-independent of Y in the context of Z if and only if they are leximin-independent and if and only if they are leximax-independent.*

3.3 Independence in Possibilistic framework

In this Section we recall well-known definitions of the independence relations which apply to a possibility distribution π [38, 39, 62, 63, 122, 129]. Clearly this distribution induces a unique plausibility relation $\geq_\pi$ using (2.11); this will enable us to compare the independence relations introduced in this section to the ones in the previous sections. The comparison results will be presented in Section 3.5.

3.3.1 Possibilistic causal independence

The idea in defining possibilistic causal independence relation based on the possibilistic conditioning is that X is considered as independent from Y in the context Z if for any instance $z \in D_Z$, the possibility degree of any $x \in D_X$ remains unchanged for any value $y \in D_Y$. More formally:

$$\Pi(x \mid y \wedge z) = \Pi(x \mid z), \forall x, y, z. \tag{3.10}$$

Since possibility theory has two kinds of conditioning (see Section 1.4.3), this leads to two definitions of causal possibilistic independence:

- **Min-based independence relation** obtained by using the min-based conditioning (1.17) in (3.10). This form of independence called I_M is not symmetric i.e. $I_M(X, Z, Y) \neq I_M(Y, Z, X)$ where Z denotes the context variable, as pointed out by Fonck [63] and as shown by the following Counter-example.

  Counter-example 3.9 *Let us consider three binary variables A, B and C with the possibility distribution given in Table[2] 3.2. We can check that $\Pi(a \mid b \wedge c) = \Pi(a \mid c), \forall a, b, c$ i.e. $I_M(A, C, B)$ is true but, $\Pi(b_1 \mid a_1 \wedge c_1) = 1 \neq \Pi(b_1 \mid c_1) = 0.7$ i.e. $I_M(B, C, A)$ is not true.*

Table 3.2: Possibility distribution on A, B and C

a	b	c	$\pi(a \wedge b \wedge c)$
a_1	-	-	0.6
a_2	b_1	c_1	0.7
a_2	b_1	c_2	0.8
a_2	b_2	c_1	0.9
a_2	b_2	c_2	1

[2] *The symbol - in the table replaces all the instances relative to the corresponding column.*

Let us denote I_{MS} the symmetrized version of I_M suggested in [62] (called MS-independence) $\forall x, y, z$:

$$\textbf{(i)}\ \Pi(x \mid_m y \wedge z) = \Pi(x \mid_m z) \text{ and}$$
$$\textbf{(ii)}\ \Pi(y \mid_m x \wedge z) = \Pi(y \mid_m z). \tag{3.11}$$

This relation is a very strong one since the MS-independence between two sets of variables X and Y requires full ignorance about one of them (uniform distribution) [38, 39] i.e.
$\Pi(x) = 1, \forall x \in D_X$ or $\Pi(y) = 1, \forall y \in D_Y$.

- **Product independence relation** obtained by using the product-based conditioning (1.18) in (3.10). We can rewrite this form of independence using:

$$\Pi(x \wedge y \mid_p z) = \Pi(x \mid_p z) \cdot \Pi(y \mid_p z), \forall x, y, z, \tag{3.12}$$

or equivalently,

$$\Pi(x \mid_p y \wedge z) = \Pi(x \mid_p z), \forall x, y, z. \tag{3.13}$$

Let us denote I_{Prod} the product based independence relation. The equivalence between (3.12) and (3.13) is true only for positive distributions. Moreover, Prod-based independence, contrary to min-based independence, is symmetric. This definition can be expressed in Spohn's ordinal function framework [119, 120] using an appropriate transformation from integers to the unit scale $[0, 1]$. Indeed, this can be checked by showing that product-based conditioning is equivalent to Spohn's conditioning [53, 56].

3.3.2 Possibilistic decompositional independence: non-interactivity

In the possibilistic framework, the standard decompositional independence between X and Y in the context Z is represented by the **non-interactivity** relation introduced by Zadeh [129], denoted by $I_{NI}(X, Z, Y)$ (NI for Non-Interactivity) and defined by:

$$\Pi(x \wedge y \mid_m z) = min(\Pi(x \mid_m z), \Pi(y \mid_m z)), \forall x, y, z. \tag{3.14}$$

Proposition 3.15 *The **non-interactivity** relation can be expressed by [62]:*

$$\Pi(x \wedge y \wedge z) = min(\Pi(x \wedge z), \Pi(y \wedge z)), \forall x, y, z. \tag{3.15}$$

The following proposition relates existing independence relations in possibility theory.

Proposition 3.16 *MS-independence relation implies I_{NI} [64] and I_{Prod} independence relations. The converse is false. However, NI and the Prod independence relations are incomparable.*

Counter-example 3.10 : I_{Prod} AND I_{NI} DO NOT IMPLY I_{MS} AND THEY ARE INCOMPARABLE

Table 3.3: Relation between I_{Prod}, I_{NI} and I_{MS}

a	b	$\pi_1(a \wedge b)$	a	b	$\pi_2(a \wedge b)$
a_1	b_1	*0.6*	a_1	b_1	*1*
a_1	b_2	*1*	a_1	b_2	*0.8*
a_2	b_1	*0.36*	a_2	b_1	*0.8*
a_2	b_2	*0.6*	a_2	b_2	*0.8*

Let us consider two binary variables A and B with the possibility distributions given in Table 3.3. We can check that in π_1, the relation $I_{Prod}(A, \emptyset, B)$ is true contrary to $I_{MS}(A, \emptyset, B)$ and $I_{NI}(A, \emptyset, B)$. Moreover in π_2, the relation $I_{NI}(A, \emptyset, B)$ is true contrary to $I_{MS}(A, \emptyset, B)$ and $I_{Prod}(A, \emptyset, B)$.

3.4 Commensurability and the decomposition of plausibility relations

The *decomposition* of a joint possibility distribution pertaining to tuples of variables into local distributions on smaller subsets of variables allows to have a reasoning machinery working at the local level without losing any information.

In the qualitative setting, forming a joint possibility relation from marginal ones is not immediate due to the absence of commensurability assumption between the different orderings. Indeed, different rankings are not expressing grades in the same scale and then it is impossible to compare the states, which makes it possible to build joint possibility relations.

This problem can be solved with the help of ranking functions which make these relations commensurable. Namely, in the possibilistic framework all the orderings are defined on the same scale e.g. [0, 1], which is very important in the recomposition of joint distributions from marginal ones.

Therefore, before decomposing the qualitative independence relations we have introduced in this chapter i.e. Pareto, leximin, leximax, POS, BPS and PT independence, we will first redefine them in the possibilistic framework as shown in subsection 3.3.1.

A natural way of defining decomposable independence relations is to analyze the structure of the possibility distribution π defined on two independent variable sets X and Y in order to

see if there is some function f such that:

$$\pi(x \wedge y) = f(\Pi(x), \Pi(y)), \forall x, y.$$

3.4.1 Decomposition of Non-Interactivity and Product independence relations

The decomposition of the non-interactivity and the Product independence relations is trivial:

- The non-interactivity relation (see (3.14)) can be defined in a qualitative setting as it is stated by the following proposition:

  Proposition 3.17 *Let π be a possibility distribution. Let $\geq_\pi$ defined by $\omega \geq_\pi \omega'$ iff $\pi(\omega) \geq \pi(\omega')$. Then, X and Y are NI-independent iff:*

  $$x \wedge y \wedge z =_\Pi x \wedge z \text{ or } x \wedge y \wedge z =_\Pi y \wedge z, \forall x, y, z. \tag{3.16}$$

  However, NI-independence is not interesting in a qualitative representation since it does not allow for the recomposition of a unique global plausibility relation from local orders defined on independent variables (due to the non-satisfaction of the commensurability property), as shown by the example below.

  Example 3.2 *Let us consider two variables, relative to climatic conditions (CCdt) and physiological accidents (PAcc), such that:*
  $D_{CCdt} = \{Bad(b), Good(g)\}$
  $D_{PAcc} = \{Yes(y), No(n)\}$ *with the following local orderings:*

  $$\textit{(i) } b >_\Pi g \textit{ and } \textit{ (ii) } y >_\Pi n.$$

  There is no unique plausibility relation $\geq_\pi$ satisfying (i) and (ii) such that (CCdt) and (PAcc) are NI-independent. Indeed, it is sufficient to consider the two plausibility relations $\geq_\pi$ and $\geq'_\pi$:

  $$b \wedge y >_\pi g \wedge y >_\pi b \wedge n =_\pi g \wedge n,$$

  $$b \wedge y >'_\pi g \wedge y ='_\pi b \wedge n ='_\pi g \wedge n.$$

  However, if the local orderings are encoded in possibility theory then we will have a unique plausibility relation $\geq_\pi$ using $\pi(ccdt \wedge pacc) = min(\pi(ccdt), \pi(pacc)), \forall ccdt \in D_{CCdt}, \forall pacc \in D_{PAcc}$.

Indeed, if we encode the plausibility relation $\geq_\pi$ by the possibility distribution given in Table 3.4, we obtain the local distributions on the two variables (CCdt) and (PAcc) given in Table 3.5. Then, from $\pi(ccdt)$ and $\pi(pacc)$ we can recover (i) and (ii) in a unique manner using the min operator.

Table 3.4: Decomposition of NI-independence

$ccdt$	$pacc$	$\pi(ccdt \wedge pacc)$
b	y	1
b	n	0.8
g	y	0.9
g	n	0.8

Table 3.5: Decomposition of NI-independence

$ccdt$	$\pi(ccdt)$	$pacc$	$\pi(pacc)$
b	1	y	1
g	0.9	n	0.8

- I_{Prod} relation enjoys the same properties as the independence relation proposed in the probabilistic framework. In particular, we can decompose it, i.e. we can recover $\Pi(x \wedge y \mid_p z)$ in a unique manner from $\Pi(x \mid_p z)$ and $\Pi(y \mid_p z)$ using the product operator.

The importance of the commensurability assumption also appears in fuzzy set based multicriteria aggregation, especially when defining connectives between fuzzy sets. For instance, French [65] questions the validity of the intersection definition of two fuzzy sets (using the minimum operator to define the membership function associated with the intersection) when no commensurability is assumed.

3.4.2 Decomposition of leximin and leximax independence relations

In the qualitative setting, even if a plausibility relation is leximin or leximax decomposable, it cannot be decomposed without loss of information again due to the absence of commensurability assumption.

Example 3.3 *Let us consider two variables, relative to climatic conditions (CCdt) and maintenance (Maint), such that:*
$D_{CCdt} = \{Bad(bc), Good(gc)\}$
$D_{Maint} = \{Good(gm), Medium(mm), Weak(wm)\}$ *with the following plausibility relation $\geq_\pi$ which is leximin decomposable:*

$$bc \wedge gm >_\pi bc \wedge gm >_\pi gc \wedge mm >_\pi bc \wedge wm >_\pi gc \wedge wm.$$

We can easily check that this plausibility relation cannot be recovered from the induced local orders on (CCdt) and (Maint) given by:

$$\textit{(i) } bc >_\Pi gc \textit{ and } \textit{ (ii) } gm >_\Pi mm >_\Pi wm.$$

Indeed, it is sufficient to consider the following plausibility relation:

$$bc \wedge gm >'_\pi gc \wedge gm >'_\pi bc \wedge mm >'_\pi gc \wedge mm >'_\pi bc \wedge wm >'_\pi gc \wedge wm,$$

which satisfies (i) and (ii) and which is also leximin-decomposable.

Such a problem can be solved when considering the scale-based setting. Indeed, in this case the decomposition of leximin and leximax decomposable distributions is immediate since we use weights represented by possibility degrees which allows the comparison between different interpretations. In other terms, if the plausibility relation $\geq_\pi$ relative to any joint possibility distribution π is leximin or leximax decomposable then we can recover π from local distributions. Without a common scale, the use of the leximin or leximax does not allow the recovering of $\geq_\pi$.

Example 3.4 *Let π be a possibility distribution encoding the plausibility relation $\geq_\pi$ given in Example 3.3 (see Table 3.6). We can recover π from the local distributions on (CCdt) and (Maint) and the numerical scale (1,.9,.8,.7,.3,.2) using the leximin ordering. Indeed, the use of the leximin on the local distributions provides the plausibility relation relative to π i.e.*

$$bc \wedge gm >_\pi bc \wedge mm >_\pi gc \wedge gm >_\pi \wedge mm >_\pi gc \wedge wm >_\pi gc \wedge wm.$$

Thus, using the numerical scale we can recover the original distribution π. For instance, the state $bc \wedge gm$ corresponds to the possibility degree 1.

Table 3.6: Decomposition of leximin-independence

$ccdt$	$maint$	$\pi(ccdt \wedge maint)$
bc	gm	1
bc	mm	0.9
bc	wm	0.3
gc	gm	0.8
gc	mm	0.7
gc	wm	0.2

3.4.3 Decomposition of POS-independence relation

In order to decompose the causal relation I_{POS}, we should see if there exists a function f such that for each possibility distribution π (encoding some plausibility relation) where X and Y are POS-independent, we have $\forall x, x' \in D_X, \forall y, y' \in D_Y$:

$$\pi(x \wedge y) > \pi(x' \wedge y') \text{ iff } f(\Pi(x), \Pi(y)) > f(\Pi(x'), \Pi(y')).$$

The following counter-example shows that this is impossible in the general case.

Counter-example 3.11 *Assume that the function* f *exists, then let us consider two variables, relative to climatic conditions (CCdt) and maintenance (Maint), such that:*
$D_{CCdt} = \{Bad(bc), Medium(mc), Good(gc)\}$
$D_{Maint} = \{Weak(wm), Medium(mm), Good(gm)\}$
with the two possibility distributions π_1 *and* π_2 *given in Table 3.7.*

Table 3.7: Decomposition of POS-independence

ccdt	*maint*	$\pi_1(ccdt \wedge maint)$	$\pi_2(ccdt \wedge maint)$
bc	*wm*	*1*	*1*
bc	*mm*	*0.9*	*0.9*
bc	*gm*	*0.5*	*0.5*
mc	*wm*	*0.8*	*0.8*
mc	*mm*	*0.3*	*0.3*
mc	*gm*	*0.2*	*0.1*
gc	*wm*	*0.4*	*0.4*
gc	*mm*	*0.1*	*0.2*
gc	*gm*	*0*	*0*

We have $\Pi_1(bc) = \Pi_2(bc) = 1, \Pi_1(mc) = \Pi_2(mc) = 0.8, \Pi_1(gc) = \Pi_2(gc) = 0.4,$
$\Pi_1(wm) = \Pi_2(wm) = 1, \Pi_1(mm) = \Pi_2(mm) = 0.9, \Pi_1(gm) = \Pi_2(gm) = 0.5.$

We can check that (CCdt) and (Maint) are POS-independent in both π_1 *and* π_2 *since* $\forall a_i, a_j \in D_{CCdt}, \forall b_k \in D_{maint}$, *we have* $\Pi_l(a_i) > \Pi_l(a_j)$ *iff* $\pi_l(a_i \wedge b_k) > \pi(a_j \wedge b_k)$,
for l = 1,2 (and the same when exchanging a and b).

Besides, π_1 *induces* $\pi_1(mc \wedge gm) > \pi_1(gc \wedge mm)$, *i.e.* $\text{f}(\Pi_1(mc), \Pi_1(gm)) > \text{f}(\Pi_1(gc), \Pi_1(mm))$
while π_2 *induces* $\pi_2(gc \wedge mm) > \pi_2(mc \wedge gm)$, *i.e.* $\text{f}(\Pi_2(gc), \Pi_2(mm)) > \text{f}(\Pi_2(mc), \Pi_2(gm))$,
hence contradiction.

However, there are particular cases where decomposition can be achieved. The first concerns binary variables, and the second two-level distributions. Indeed, in these two cases

POS-independent is equivalent to leximin and leximax independence relations (see Propositions 3.13 and 3.14 then it can be decomposed using these two operators.
Note that the decomposition of two-level distributions can be useful for databases when the existing tuples are preferred to absent ones.

POS-independence is a weak relation in the perspective of decomposition. The following proposition makes the weakness of POS-independence explicit:

Proposition 3.18 *X and Y are POS-independent in $\geq_\pi$ iff $\geq_\pi$ is Pareto-compatible (i.e., monotonic) on X and Y.*

3.5 Comparative study

In Sections 3.2 and 3.3, we have established the different links existing, on the one hand, between scale-based possibilistic independence relations and, in the other hand, between independence relations expressed from plausibility relations. This Section compares all these relations. Namely, given a joint possibility distribution π, we will relate the relations I_{Pareto}, I_{POS}, $I_{leximin}$ and $I_{leximax}$ of Section 3.2 to the ones of Section 3.3 (i.e. I_{NI}, I_{MS} and I_{Prod}) by considering the plausibility relation $\geq_\pi$ induced from π by using (2.11).

Using Proposition 3.10, we can show the equivalence between MS and Pareto independence relations.

Proposition 3.19 *Let π be a possibility distribution, and $\geq_\pi$ be its associated plausibility relation. Then, X and Y are MS-independent in π if and only if they are Pareto-independent in $\geq_\pi$.*

Proposition 3.20, shows that the M-independence implies the PO-independence.

Proposition 3.20 *If X is M-independent of Y in the context Z, then X is also PO-independent of Y in the same context. The converse is not true.*

Counter-example 3.12 *: I_{PO}* DOES NOT IMPLY *I_M*

Let us consider two binary variables A and B with the possibility distribution given in Table 3.8. We can check that the relation $I_{PO}(A, \emptyset, B)$ is true contrary to $I_M(A, \emptyset, B)$.

Propositions 3.21 and counter-example 3.15 relate Prod-independence to POS, leximin and leximax independencies.

Table 3.8: Relation between I_M and I_{PO}

a	b	$\pi(a \wedge b)$
a_1	b_1	*1*
a_1	b_2	*0.8*
a_2	b_1	*0.5*
a_2	b_2	*0.7*

Proposition 3.21 *If X and Y are Prod-independent in a strictly positive possibility distribution π, then X is POS-independent of Y in the plausibility relation induced by π. The converse is false.*

Counter-example 3.13 *: I_{POS}* DOES NOT IMPLY *I_{Prod}*

Table 3.9: Relation between I_{POS} and I_{Prod}

a	b	$\pi(a \wedge b)$	a	b	$\pi(a \wedge b)$
a_1	b_1	*1*	a_2	b_1	*0.8*
a_1	b_2	*0.9*	a_2	b_2	*0.7*
a_1	b_3	*0.6*	a_2	b_3	*0.5*

Let A and B be two variables with the strictly positive possibility distribution given in Table 3.9. We can check that the relation $I_{POS}(A, \emptyset, B)$ is true contrary to $I_{Prod}(A, \emptyset, B)$ since $\pi(a_2 \wedge b_3) = 0.5 \neq \Pi(a_2) \cdot \Pi(b_3) = 0.48$.

Counter-example 3.14 *I_{Prod}* IMPLIES *I_{POS}* ONLY WITH STRICTLY POSITIVE DISTRIBUTIONS

Proposition 3.21 is false for non strictly positive possibility distributions. Indeed, let us consider two binary variables A and B with the non strictly positive possibility distribution given in Table 3.10. We can check that $I_{Prod}(A, \emptyset, B)$ is true, contrary to $I_{POS}(A, \emptyset, B)$.

Table 3.10: Relation between I_{Prod} and I_{POS}

a	b	$\pi(a \wedge b)$
a_1	b_1	*1*
a_1	b_2	*0*
a_2	b_1	*0*
a_2	b_2	*0*

In the general case, the leximin and leximax independencies are incomparable with Prod-independence as shown by the following counter-example:

Table 3.11: Relation between I_{Prod}, $I_{leximin}$ and $I_{leximax}$

a	b	$\pi(a \wedge b)$	a	b	$\pi(a \wedge b)$	a	b	$\pi(a \wedge b)$
a_1	b_1	*1*	a_2	b_1	*0.5*	a_3	b_1	*0.4*
a_1	b_2	*0.8*	a_2	b_2	*0.4*	a_3	b_2	*0.32*

Table 3.12: Relation between I_{Prod}, $I_{leximin}$ and $I_{leximax}$

a	b	$\pi(a \wedge b)$	a	b	$\pi(a \wedge b)$
a_1	b_1	*1*	a_2	b_1	*0.8*
a_1	b_2	*0.9*	a_2	b_2	*0.3*
a_1	b_3	*0.5*	a_2	b_3	*0.2*

Counter-example 3.15 *:* I_{Prod} INDEPENDENCE IMPLIES NEITHER $I_{leximin}$ NOR $I_{leximax}$ AND VICE VERSA

Let A and B be two variables,

- *with the possibility distribution given in Table 3.11, we can check that $I_{Prod}(A, \emptyset, B)$ is satisfied while $I_{leximin}(A, \emptyset, B)$ and $I_{leximax}(A, \emptyset, B)$ are false. Note that the product operator allows for compensation contrary to the leximin and leximax orderings. For instance, if we have two pairs $(\Pi(x), \Pi(y))$ and $(\Pi(x'), \Pi(y'))$ such that $\Pi(x) > \Pi(x')$ and $\Pi(y') > \Pi(y)$, then $x \wedge y$ and $x' \wedge y'$ will be always strictly ranked using leximin and leximax principle. However, they can be equally ranked using the product operator since it may happen that $\Pi(x) \cdot \Pi(y) = \Pi(x') \cdot \Pi(y')$. This is the case in this example since $\Pi(a_2) = 0.5 > \Pi(a_3) = 0.4$ and $\Pi(b_1) = 1 > \Pi(b_2) = 0.8$ and $\Pi(a_2) \cdot \Pi(b_2) = \Pi(a_3) \cdot \Pi(b_1) = 0.4$.*

- *with the possibility distributions given in Table 3.12, we can check that $I_{leximax}(A, \emptyset, B)$ is respected while $I_{Prod}(A, \emptyset, B)$ is false since $\pi(a_2 \wedge b_2) = 0.3 \neq \Pi(a_2) \cdot \Pi(b_2) = 0.72$. Moreover, with the possibility distribution given in Table 3.6, we can check that $I_{leximin}(A, \emptyset, B)$ is respected contrary to $I_{Prod}(A, \emptyset, B)$ since $\pi(a_2 \wedge b_3) = 0.5 \neq \Pi(a_2) \cdot \Pi(b_3) = 0.48$.*

Proposition 3.22 relates NI-independence to independence relations defined on plausibility relations.

Proposition 3.22 *Pareto independence implies NI-independence relation (since Pareto is equivalent to MS). Moreover, NI-independence implies PT-independence. However, this independence relation is incomparable with the other qualitative independence relations, namely the leximin, leximax, POS and BPS independencies.*

Counter-example 3.16 *:* I_{NI} IS INCOMPARABLE WITH $I_{leximin}$, $I_{leximax}$, I_{POS} AND I_{BPS}

In the possibility distribution π_2 given in Table 3.3, we can check that $I_{NI}(A, \emptyset, B)$ is true. However in the plausibility relation induced by π_2 (i.e., $a_1 \wedge b_1 >_\pi a_1 \wedge b_2 =_\pi a_2 \wedge b_1 =_\pi a_2 \wedge b_2$) the relations $I_{leximin}(A, \emptyset, B)$ and $I_{leximax}(A, \emptyset, B)$ are false since $a_1 \wedge b_2 =_\pi a_2 \wedge b_2$ while $max(a_1, b_2) >_\Pi max(a_2, b_2)$. Moreover, $I_{POS}(A, \emptyset, B)$ is false since the local plausibility relation relative to A is $a_1 >_\Pi a_2$ while $a_1 =_\Pi a_2$ in the context of b_1. Lastly, $I_{BPS}(A, \emptyset, B)$ is false since $\mathbf{Acc}(a_1 \mid b_2) = 0 \neq \mathbf{Acc}(a_1) = 1$.

In Table 3.12, we can check that the relation $I_{leximax}(A, \emptyset, B)$ is respected contrary to $I_{NI}(A, \emptyset, B)$ since $\pi(a_2 \wedge b_2) = 0.3 \neq min(\Pi(a_2), \Pi(b_2)) = min(0.8, 0.9) = 0.8$.

Lastly, in Table 3.9, we can check that $I_{leximin}(A, \emptyset, B)$ is respected contrary to $I_{NI}(A, \emptyset, B)$ since $\pi(a_2 \wedge b_2) = 0.7 \neq min(\Pi(a_2), \Pi(b_2)) = min(0.8, 0.9) = 0.8$. Then we can deduce that in π the relations $I_{POS}(A, \emptyset, B)$, $I_{BPS}(A, \emptyset, B)$ and $I_{PT}(A, \emptyset, B)$ are true contrary to NI-independence since $I_{leximin}$ implies these three independence relations (from Propositions 3.9, 3.4 and 3.11).

This counter-example shows that if we start with a complete order, and map it to a scale (e.g. $[0, 1]$) then if X and Y are NI-independent (which implies that the distribution is decomposable with the minimum operator), then it is always possible to recompose the initial ordering from local ones defined on X and Y. This is possible because we can store the total pre-order by mapping it to a totally ordered scale. However, the case where the commensurability is crucial is when the expert provides local orders on X and Y and the fact that these sets of variables are NI-independent. Then, it is no longer possible to construct the global distribution.

Figure 3.1: Links between symmetric (a) and non-symmetric (b) independence relations

Figure 3.1 (a) illustrates the links existing between the different symmetric independence relations. Figure 3.1 (b) concerns non-symmetric independence relations. The arrows show the inclusion between independence relations (transitivity is not explicit for sake of clarity). The absence of arrows implies the incomparability of the independence relations. Note that I_{MS} and I_{Pareto} are the strongest independence relations since MS (or equivalently, Pareto) independence between two sets of variables imply a lack of information on one of them. However, I_{PT} is the weakest one, since it is sufficient to satisfy any independence relation in order

to confirm that this relation is true as stated by Proposition 3.7. Finally, we remark that I_{NI} is implied by I_{MS} and I_{Pareto} and that it implies I_{PT} but it is incomparable with the other independence relations. Unsurprisingly, there are several decompositional independence relations according to the chosen decomposition mode. However, there is only one natural causality-oriented independence which is POS-independence relation. Fortunately, all of decompositional independence relations (except I_{NI}) are also meaningful from a causality point of view.

3.6 Graphoid properties

The independence relations can be characterized by some properties which have been initially established for probabilistic conditional independence [25, 37, 103, 104]. These properties are as follows:

- *P1: Symmetry* : $I(X, Z, Y) \Rightarrow I(Y, Z, X)$
  This relation asserts that in any state of knowledge Z, if Y tells us nothing new about Y, then X tells us nothing new about Y.

- *P2: Decomposition*: $I(X, Z, Y \cup W) \Rightarrow I(X, Z, Y)$ and $I(X, Z, W)$
  This relation asserts that if (Y and W) are irrelevant to X then Y (resp. W) is irrelevant to X.

- *P3: Weak union*: $I(X, Z, Y \cup W) \Rightarrow I(X, Z \cup Y, W)$
  This relation asserts that the learning of an irrelevant information W, cannot transform an irrelevant information Y into a pertinent one for X.

- *P4: Contraction*: $I(X, Z \cup Y, W)$ and $I(X, Z, Y) \Rightarrow I(X, Z, Y \cup W)$
  This relation asserts that if W is irrelevant to X after receiving irrelevant information Y, then W should be also irrelevant to X knowing Y. Together, the weak union and this property state that irrelevant information should not modify the relevance of other propositions.

- *P5: Intersection*: $I(X, Z \cup W, Y)$ and $I(X, Z \cup Y, W) \Rightarrow I(X, Z, Y \cup W)$
  This relation states that if Y is irrelevant to X when W is known and if W is irrelevant to X when Y is known, then neither W, nor Y, nor their combination is relevant to X.

Any independence structure that satisfies the properties P1-P4 is called a *semi-graphoid*. If it also satisfies property P5 it is said to be a *graphoid*. It has been shown that the probabilistic independence relation is a *semi-graphoid*, and it is a *graphoid* if the considered probability

distribution is strictly positive (i.e. $p > 0$) [103].

We first recall existing results on graphoid properties in possibilistic framework. Indeed, Fonck [63] has shown that I_{NI} and I_{Prod} relations are *semi-graphoids*. Indeed I_{NI} does not satisfy the intersection property, while I_{Prod} satisfies this property only if we consider strictly positive distributions. I_M independence relation satisfies all *graphoid* properties except the symmetry and its symmetrized version I_{MS} is a *graphoid*. This implies that I_{Pareto} is a *graphoid* too since these two relations are equivalent as stated by Proposition 3.19.

The following propositions establish the graphoid properties of proposed independence relations.

Proposition 3.23 *I_{PO} independence relation satisfies all graphoid properties except the symmetry.*

For the lack of symmetry see Counter-example 3.3. Note the unexpected result: the addition of symmetry to I_{PO} leads to the loss of contraction and intersection properties.

Proposition 3.24 *I_{POS} relation is not a semi-graphoid, since it satisfies the symmetry (by definition), the decomposition and the weak union but neither the contraction nor the intersection properties.*

Counter-example 3.17 *:* Lack of contraction property for I_{POS}

Let us consider three binary variables A, B and C with the following plausibility relation: $a_2 \wedge b_2 \wedge c_1 >_\pi a_2 \wedge b_2 \wedge c_2 >_\pi a_2 \wedge b_1 \wedge c_1 >_\pi a_2 \wedge b_1 \wedge c_2 >_\pi a_1 \wedge b_2 \wedge c_1 >_\pi a_1 \wedge b_1 \wedge c_1 >_\pi a_1 \wedge b_2 \wedge c_2 >_\pi a_1 \wedge b_1 \wedge c_2$.

It can be checked that:

$$I_{POS}(A,B,C) \text{ is true since } \begin{cases} \forall b,c,\forall a,a' \in D_A, a \wedge b >_\Pi a' \wedge b \text{ iff } a \wedge b \wedge c >_\pi a' \wedge b \wedge c \text{ and} \\ \forall a,b,\forall c,c' \in D_C, c \wedge b >_\Pi c' \wedge b \text{ iff } c \wedge a \wedge b >_\pi c' \wedge a \wedge b \end{cases}$$

$$I_{POS}(A,\emptyset,B) \text{ is true since } \begin{cases} \forall b,\forall a,a' \in D_A, a >_\Pi a' \text{ iff } a \wedge b >_\Pi a' \wedge b \text{ and} \\ \forall a,\forall b,b' \in D_B, b >_\Pi b' \text{ iff } b \wedge a >_\Pi b' \wedge a \end{cases}$$

However, $I_{POS}(A,\emptyset,B \cup C)$ is not verified since $b_2 \wedge c_2 >_\Pi b_1 \wedge c_1$ while in the context a_1, we have $b_1 \wedge c_1 >_\Pi b_2 \wedge c_2$.

Counter-example 3.18 *:* Lack of intersection property for I_{POS}

Let us consider the plausibility relation given in Counter-example 3.17. It can be checked that:

$$I_{POS}(A,C,B) \text{ is true since } \begin{cases} \forall b,c,\forall a,a' \in D_A, a \wedge c >_\Pi a' \wedge c \text{ iff } a \wedge b \wedge c >_\pi a' \wedge b \wedge c \text{ and} \\ \forall a,c,\forall b,b' \in D_B, b \wedge c >_\Pi b' \wedge c \text{ iff } b \wedge a \wedge c >_\pi b' \wedge a \wedge c \end{cases}$$

$I_{POS}(A,B,C)$ *is true since* $\begin{cases} \forall b,c,\forall a,a' \in D_A, a\wedge b >_\Pi a'\wedge b \text{ iff } a\wedge b\wedge c >_\pi a'\wedge b\wedge c \text{ and} \\ \forall a,b,\forall c,c' \in D_C, c\wedge b >_\Pi c'\wedge b \text{ iff } c\wedge a\wedge b >_\pi c'\wedge a\wedge b \end{cases}$
However, $I_{POS}(A,\emptyset,B\cup C)$ *is not verified since* $b_2\wedge c_2 >_\Pi b_1\wedge c_1$ *while in the context* a_1*, we have* $b_1\wedge c_1 >_\Pi b_2\wedge c_2$.

Proposition 3.25 *$I_{leximax}$ relation only satisfies the symmetry and the decomposition but not the other graphoid properties i.e. weak union, contraction and intersection. $I_{leximin}$ relation satisfies the same properties than $I_{leximax}$.*

Some properties may be recovered in particular cases. For instance in the case of two-level distributions, $\mathrm{I}_{leximax}$ and $\mathrm{I}_{leximin}$ relations satisfy the weak union since they are equivalent to I_{POS} (see Propositions 3.13 and 3.14).

Counter-example 3.19 *:* LACK OF WEAK UNION PROPERTY FOR $I_{leximax}$

Let us consider three variables A, B and C with the following plausibility relation:
$a_1\wedge b_1\wedge c_1 >_\pi a_2\wedge b_1\wedge c_1 >_\pi a_1\wedge b_2\wedge c_2 >_\pi a_3\wedge b_1\wedge c_1 =_\pi a_1\wedge b_1\wedge c_2 >_\pi a_1\wedge b_2\wedge c_1 >_\pi a_2\wedge b_2\wedge c_2 >_\pi a_2\wedge b_1\wedge c_2 >_\pi a_2\wedge b_2\wedge c_1 >_\pi a_3\wedge b_2\wedge c_2 >_\pi a_3\wedge b_1\wedge c_2 >_\pi a_3\wedge b_2\wedge c_1$.
It can be checked that $I_{leximax}(A,\emptyset,B\cup C)$ *is true since* $a_1 =_\Pi b_1\wedge c_1 >_\Pi a_2 >_\Pi a_3 =_\Pi b_1\wedge c_2 >_\Pi b_2\wedge c_1 >_\Pi b_2\wedge c_2$*. However,* $I_{leximax}(A,B,C)$ *is false since* $a_2\wedge b_2\wedge c_1 >_\Pi a_3\wedge b_2\wedge c_2$ *while* $max(a_3\wedge b_2, b_2\wedge c_2) >_\Pi max(a_2\wedge b_2, b_2\wedge c_1)$.

Counter-example 3.20 *:* LACK OF CONTRACTION PROPERTY FOR $I_{leximax}$

Let us consider three binary variables A, B and C with the following plausibility relation:
$a_1\wedge b_2\wedge c_1 >_\pi a_1\wedge b_2\wedge c_2 =_\pi a_2\wedge b_2\wedge c_1 >_\pi a_2\wedge b_2\wedge c_2 >_\pi a_1\wedge b_1\wedge c_2 >_\pi a_1\wedge b_1\wedge c_1 =_\pi a_2\wedge b_1\wedge c_2 >_\pi a_2\wedge b_1\wedge c_1$.
It can be checked that $I_{leximax}(A,\emptyset,B)$ *and* $I_{leximax}(A,B,C)$ *are true since* $a_1 =_\Pi b_2 >_\Pi a_2 >_\Pi b_1$ *and* $a_1\wedge b_2 =_\Pi b_2\wedge c_1 >_\Pi a_2\wedge b_2 =_\Pi b_2\wedge c_2 >_\Pi a_1\wedge b_1 =_\Pi b_1\wedge c_2 >_\Pi a_2\wedge b_1 =_\Pi b_1\wedge c_1$.
However, $I_{leximax}(A,\emptyset,B\cup C)$ *is false since* $a_1\wedge b_1\wedge c_1 =_\pi a_2\wedge b_1\wedge c_2$ *while* $max(a_1, b_1\wedge c_1) >_\Pi max(a_2, b_1\wedge c_2)$.

Counter-example 3.21 *:* LACK OF INTERSECTION PROPERTY FOR $I_{leximax}$

Let us consider three binary variables A, B and C with the following plausibility relation:
$a_1\wedge b_1\wedge c_1 =_\pi a_1\wedge b_2\wedge c_2 >_\pi a_1\wedge b_2\wedge c_1 >_\pi a_1\wedge b_1\wedge c_2 >_\pi a_2\wedge b_2\wedge c_2 >_\pi a_2\wedge b_1\wedge c_1 >_\pi a_2\wedge b_2\wedge c_1 >_\pi a_2\wedge b_1\wedge c_2$.
It can be checked that $I_{leximax}(A,C,B)$ *and* $I_{leximax}(A,B,C)$ *are true since* $a_1\wedge c_1 =_\Pi a_1\wedge c_2 =_\Pi b_1\wedge c_1 =_\Pi b_2\wedge c_2 >_\Pi b_2\wedge c_1 >_\Pi b_1\wedge c_2 >_\Pi a_2\wedge c_2 >_\Pi a_2\wedge c_1$ *and* $a_1\wedge b_1 =_\Pi a_1\wedge b_2 =_\Pi b_1\wedge c_1 =_\Pi b_2\wedge c_2 >_\Pi b_2\wedge c_1 >_\Pi b_1\wedge c_2 >_\Pi a_2\wedge b_2 >_\Pi a_2\wedge b_1$*. However,* $I_{leximax}(A,\emptyset,B\cup C)$ *is false since* $a_2\wedge b_2\wedge c_2 >_\Pi a_2\wedge b_1\wedge c_1$ *while* $max(a_2, b_2\wedge c_2) =_\Pi max(a_2, b_1\wedge c_1)$ *and* $min(a_2, b_2\wedge c_2) =_\Pi min(a_2, b_1\wedge c_1)$.

The main results regarding graphoid properties are summarized by Table 3.13 where we have grouped the established properties concerning I_{NI}, I_M, I_{MS}, and I_{Prod} and the new ones concerning $I_{Pareto}, I_{PO}, I_{POS}, I_{leximax}$ and $I_{leximin}$.

Table 3.13: Graphoid properties

Relation	Symmetry	Decomposition	Weak union	Contraction	Intersection
I_{NI}	yes	yes	yes	yes	no
I_M	no	yes	yes	yes	yes
I_{MS}	yes	yes	yes	yes	yes
I_{Prod}	yes	yes	yes	yes	yes if $\pi > 0$
I_{Pareto}	yes	yes	yes	yes	yes
I_{PO}	no	yes	yes	yes	yes
I_{POS}	yes	yes	yes	no	no
$I_{leximax}$	yes	yes	no	no	no
$I_{leximin}$	yes	yes	no	no	no

Note that only the decompositional independence relations based on the product and the minimum operator (i.e. I_{Prod} and I_{NI}) are "*reasonable*" relations with good properties since they are semi-graphoids. Indeed, the min-based causal independence relation I_{MS} has good properties but is too strong (see Figure 3.1) to be practically used.

The property of semi-graphoid is crucial when developing graphical approaches in uncertain reasoning as we will detail in Part II of this work.

In Part III we exploit the fact that even if the NI-independence relation is not a graphoid (lack of intersection) it satisfies the specific property of *idempotence* of the minimum operator which is useful for simplifying the calculus during the propagation process.

3.7 Software for testing independence relations

In this section, we propose a software implementing the possibilistic and qualitative independence relations studied in this Chapter. This software uses Matlab 6.0. and allows to test independence relations satisfied by any possibility distribution or plausibility relation. It can be incorporated in other softwares as a module for testing independence relations.
Figure 3.2 is relative to qualitative independence relations, it allows to test:

- Belief-preserving independence,
- Belief-preserving independence in its symmetrized version,

Figure 3.2: Qualitative independence relations

Figure 3.3: Possibilistic independence relations

- Order-preserving independence,
- Order-preserving independence in its symmetrized version,
- Belief decompositional independence,
- Pareto independence,
- Leximin independence,
- Leximax independence.

Figure 3.3 is relative to possibilistic independence relations, it allows to test:

- Min-based independence,
- Min-based independence in its symmetrized version,
- Product-based independence,
- Non-interactivity.

Once one of these independence relations is selected, we should define the number of variables and their cardinalities (the size of their domains). Then we should define the plausibility relation or the possibility relation relative to these variables. Finally, we should specify the sets X, Y and Z and the software tests the validity of the chosen independence relation between X and Y in the context Z.

The following dialog boxes are relative to the independence relation studied in Counter-example 3.5. Indeed, Figure 3.4 defines the number of variables and their cardinalities (i.e. two binary variables). Then, Figure 3.5 defines the plausibility relation (i.e. $a_1 \wedge b_2 >_\pi a_2 \wedge b_2 >_\pi a_1 \wedge b_1 =_\pi a_2 \wedge b_1$). Finally Figure 3.6 specifies the sets X, Y and Z (i.e. $X = \{A\}$, $Y = \{B\}$ and $Z = \emptyset$) and 3.7 displays the result (i.e. the relation $I_{PT}(A, \emptyset, B)$ is true).

Figure 3.4: Definition of the number of variables and their cardinalities

Figure 3.5: Definition of the plausibility relation

3.8 Conclusion

In this Chapter, we have defined the notion of qualitative independence, and related it to the basic existing independence relations in possibility theory. Two kinds of independence have been investigated: *causal* and *decompositional* ones. We can observe that the independence relations which can be used for the decomposition of a joint distribution in possibility theory is not unique, contrary to probability theory where only the product-based independence is used. In possibility theory, alternative operators-based independence, like leximin or leximax, can be used as well.

The notions of independence proposed in this Chapter extend previous works in default reasoning [14], and belief revision [45] on independence between events to the case of variables which are not necessarily binary.

We have shown that the use of common scale (for instance the interval $[0, 1]$) is important for decomposing distributions. In addition, we have provided a comparative study between already known definitions of possibilistic independence and the ones proposed in this Chapter. This study, have shown that all of decompositional independence relations (except I_{NI}) are meaningful from a causality point of view.

We have also studied the graphoid properties of proposed independence relations.

Lastly, we have proposed a allowing to test independence relations satisfied by any possibility distribution or plausibility relation.

Results of this Chapter will be used for designing possibilistic counterparts of local propagation algorithms in causal networks. This is the aim of Parts II and III of this thesis.

Figure 3.6: Definition of X, Y and Z (here $Z = \emptyset$)

Figure 3.7: Result of testing $I(X, Z, Y)$

Part II

Possibilistic Adaptation of Probabilistic Causal Networks

Introduction Part II

Directed causal possibilistic networks, which are possibilistic counterparts of Bayesian networks, present a new promising area of research. Indeed, they offer a natural way to handle structured knowledge, with non-binary variables, using independence relations between variables.

Existing works on possibilistic graphical models are either a direct adaptation of probabilistic approach or a way to perform learning from imprecise data.

In possibility theory there are two different ways to define the counterpart of Bayesian networks. This is due to the existence of two definitions of possibilistic conditioning: *product-based* and *min-based* conditioning [38, 39, 54, 78]. In the rest of this thesis we investigate these two kinds of networks.

In this part, we first present probabilistic Bayesian networks and their propagation algorithms (Chapter 4). Then, we propose possibilistic counterparts of Bayesian networks based on the *minimum* and the *product* operators and discuss the *coherence* problem in the case of min-based possibilistic networks (Chapter 5). Finally, we propose a possibilistic adaptation of exact probabilistic propagation algorithms for product-based and min-based possibilistic networks (Chapter 6).

Chapter 4

Probabilistic Graphical Models

4.1 Introduction

Graphical models are knowledge representation tools commonly used by an increasing number of researchers. Most of graphical models refer to probability theory. Namely, *Bayesian networks* [84, 103, 94], *decision trees* [106], *Influence diagrams* [107, 108, 81] and a more general framework is *Valuation Based Systems* (VBS) [111, 112, 113, 114, 115].

Bayesian networks are used in several real world applications. For instance, they have been implemented in some of Microsoft's products. One of the most famous applications is the one used by LUMIERE project which focused on the construction and integration of Bayesian models of a user's needs for assistance. LUMIERE research led to the Office Assistant, a Bayesian help system in Office'97 [80]. Moreover, Bayesian networks are also used in industry. For instance, in [76] an expert diagnostic system (FIXIT) has been proposed for liberating users from burdensome information-retrieval activities while incurring minimal system-development and runtime costs. Several applications of Bayesian networks in the medical field have also been proposed. We can mention the QMR-DT (Quick Medical Reference Decision Theoretic) network which is a two-level graphical model where the top level of the graph contains nodes for the diseases, and the bottom level contains nodes for the findings [77, 83].

In a graphical model, we can distinguish two components, *i)* a *graphical component* which can be a directed or undirected graph, where nodes represent variables characterizing the given domain and links (edges or arcs) the causal relations between these variables. This graphical component can be provided by experts or learned from data bases, *ii)* a *numerical component* which quantifies different links and corresponds to a numerical representation of uncertainty

depending on the graph structure.

In this Chapter, we are, in particular, interested in *Bayesian networks* which are Directed Acyclic Graphs where uncertainty is encoded with conditional probabilities. We address the propagation problem consisting in updating the network when new observations arrive.

Section 4.2, presents the basic notions and notations about DAGs. Section 4.4 introduces *Bayesian networks*. Then, Section 4.6 presents Pearl's propagation algorithm in singly connected DAGs [103, 105] and Section 4.7 is dedicated to the more general case of propagation in multiply connected DAGs [84]. Finally, Section 4.8, gives a brief presentation of Valuation Based Systems (VBS) which are more general graphical models proposed by Shenoy [111, 112, 113, 114, 115].

4.2 Background and notations on graphs

Let $V = \{A, B, C, ...\}$ be a finite set of variables. Let E be a part of $V \times V$. Then, $\mathcal{G} = (V, E)$ is said to be a **graph** on V and E corresponds to the set of **edges** connecting some pairs of nodes in V. If the edges in E are oriented then they are called **arcs** and $\mathcal{G} = (V, E)$ is said to be a **directed graph**.

- for each arc AB, the node A is called its origin and B its end.
- in an arc AB, the node A is the **parent** of B and the node B is the **child** of A,
- a **root** is a node with no parents (in Figure 4.1(a) A is a root),
- a **leaf** is a node with no children (in Figure 4.1(a) B and D are leaves),
- two nodes linked by an edge (resp. arc) are said to be **adjacent**,
- a **path** in a directed graph is a sequence of nodes from one node to another using the arcs,
- a **chain** in a graph is a sequence of nodes from one node to another using the edges (in Figure 4.1(c) ACB is a path),
- a **cycle** is a path visiting each node once and having the same first and last node (in Figure 4.1(b) ACDBA is a cycle),
- a **loop** is an *undirected* cycle (in Figure 4.1(c) $ABCA$ is a loop),

- a **DAG** is a Directed Acyclic (without cycles) Graph (Figures 3.1(a) and (b) are DAGs while Figure 4.1(b) is not a DAG since it contains the circuit ABDCA).
- A **singly connected DAG or polytree** is a DAG which contains no loops, in this case the graph obtained by dropping the directions of the links is a tree (see Figure 4.1(a)).
- A **multiply connected DAG** is a DAG which can contains loops (see Figure 4.1(c) which contains a loop ABCA).

For any node $A \in V$ of a DAG $\mathcal{G}$ corresponds the following sets:

- U_A is the **parent** set of A,
- X_A the **descendant** set of A, such that there is a path from A to each $X_i \in X_A$,
- Y_A the **child** set of A,
- Z_A the **non-descendant** set of A, such that $Z_A = V - X_A$.

Figure 4.1: Examples of oriented graphs

When there is no ambiguity we use U, X, Y and Z. We use a, u_A, x_A, y_A and z_A to denote, respectively, possible instances of A, U_A, X_A, Y_A and Z_A. When there is no ambiguity we use u, x, y and z.

Example 4.1 *Let us consider the node A in the DAG of Figure 4.2, then we have:* $U_A = \{B, C\}$, $X_A = \{G, H, J, K, l, M\}$, $Y_A = \{G, H\}$ *and* $Z_A = \{B, C, D, E, F, I\}$.

Figure 4.2: Example of DAG

4.3 Conditional independence in DAGs: d-separation

The DAG structure encodes independence relations without taking into account any numerical values evaluating its links. Pearl, Verma and Geiger [70, 71, 103, 125] have investigated the problem of determining exactly what independencies are implied by the DAG structure in a causal network. The main results are presented in the following subsections:

4.3.1 DAGs topology

Given two variables A and $B \in V$ in a DAG $\mathcal{G}= (V, E)$, if there exists an arc (A, B) between A and B, then the dependency relation is evident. However, if there exists a unique chain between A and B containing an intermediary variable C, then we can face one of the following three configurations:

Figure 4.3: DAGs topology

- **Head-to-head**: if we know the value of C, then A can give a supplementary information on B and vice versa, so A and B are *not independent* in the context C (see Figure 4.3(a)). For example, Physiological accidents (A), are independent of fire (B). However, if the land yield decreases (C), we can think that it is the effect of any physiological accident (A). Nevertheless, if we receive an information about several successive fires in the same region, our belief about physiological accidents (A) decreases.

- **Head-to-tail**: if we know the value of C, then A will be irrelevant to C and consequently to B, so A and B are *independent* in the context C (see Figure 4.3(b)).
  For example, if the climatic conditions (A) are favorable to the corn culture, then the production will be plentiful. Moreover, if the production is plentiful (C), then the corn price (B) will decrease. If we know that the production is plentiful (C), then any information about climatic conditions (A) is irrelevant to the corn production (C).

- **Tail-to-tail**: if we know the value of C, then A will be irrelevant to B and vice versa, so A and B are *independent* in the context C (see Figure 4.3(c)).
  For example, if the grass of my garden is wet (A), I can think that it rained yesterday (C) and that the grass of my neighbor garden is wet too (B). However, if I'm sure that it rained yesterday (C), I can claim that the grass of my neighbor garden is wet (B) and any information about the state of the grass in my garden (A) has no influence.

4.3.2 d-separation criterion

The three cases presented above, enable us to determine, which nodes are concerned by the arrival of any observation. This can be very helpful, especially for updating networks having complex structure.

Definition 4.1 *Let $\mathcal{G} = (V, E)$ be a DAG, $Z \subseteq V$ and let A and B be two nodes in $V - Z$. Then, a chain ρ, between A and B is said to be* ***blocked*** *by Z if we are in one of the following situations:*

- ***case 1****: there is a node $Z_i \in Z$ on the chain ρ, such that the arcs which determine that Z_i is in ρ meet* ***tail-to-tail*** *at Z_i.*
- ***case 2****: there is a node $Z_i \in Z$ on the chain ρ, such that the arcs which determine that Z_i is in ρ meet* ***head-to-tail*** *at Z_i.*
- ***case 3****: there is a node $Z_i \in V$ on the chain ρ such that:*
  Z_i and none of its dependents are in Z, and
  the arcs which determine that Z_i is in ρ meet ***head-to-head*** *at Z_i.*

Using this definition, we can define a general rule describing where an information is blocked or not, known as the *d-separation* criterion:

Definition 4.2 d-separation*:*
i) Let $\mathcal{G} = (V, E)$ be a DAG, $Z \subseteq V$ and let A and B be two nodes in $V - Z$. Then, A and B are said to be ***d-separated*** *by Z if every chain between A and B is blocked by Z. We denote this property by: $< A \mid Z \mid B >_{\mathcal{G}}$.*
ii) Let $\mathcal{G} = (V, E)$ be a DAG, and let X, Y and Z be disjoint subsets of V. Then, X and Y are said to be ***d-separated*** *by Z if for every node $X_i \in X$ and $Y_i \in Y$, X_i and Y_i are d-separated by Z.*

Example 4.2 *In the DAG of Figure 4.4 we can detect the following d-separation relations:*

- *the chains $FBAC$ and $FGEC$ are blocked by A, since:*
  - the chain $FBAC$ contains the node A such that the arcs AB and AC meet tail-to-tail to A,
  - the chain $FGEC$ contains the node $\mathcal{G}$ such that the arcs FG and EG meet head-to-head at $\mathcal{G}$, in addition, neither $\mathcal{G}$ nor its descendants ($\{H\}$) are in A (see Figure 4.4(a)).
  $\Rightarrow < \{F\} \mid \{A\} \mid \{C\} >_{\mathcal{G}}$.
- *all the chains starting from $\{A, B, D\}$ to $\{E, G, H\}$ meet head-to-tail the set $\{F, C\}$ (see Figure 4.4(b)).*
  $\Rightarrow < \{A, B, D\} \mid \{F, C\} \mid \{E, G, H\} >_{\mathcal{G}}$.

Figure 4.4: Example of d-separation

4.4 Bayesian networks

The following introduces the definition of Bayesian networks when a joint distribution is available

Definition 4.3 *Let p be a joint probability distribution on the set V, and $\mathcal{G} = (V, E)$ be a DAG. ($\mathcal{G}$,p) is said to be a Bayesian network if each variable $A \in V$ is conditionally independent of* its non-descendants (Z_A) given its parents (U_A).

However, in practice, a numerical representation of $p(A_1, .., A_N)$ is rarely available, thus we should use local distributions on each node.

A Bayesian network over a set of variables V consists of two components:
- A *graphical component* composed of a directed acyclic graph (DAG) $\mathcal{G}$ reflecting the causal relations relative to the modeled domain.
- A *numerical component* consisting in a quantification of different links in the DAG by a **conditional** probability distribution of each node A in the context of its parents (U_A). Such conditional probabilities should respect the following normalization constraints for each variable A:

- if $U_A = \emptyset$ (i.e. A is a root), then the a priori probability relative to A should satisfy:

$$\sum_a P(a) = 1,$$

- if $U_A \neq \emptyset$, then the relative conditional probability relative to A in the context of its parents U_A should satisfy:

$$\sum_a P(a \mid u_A) = 1.$$

4.4.1 Probabilistic chain rule

Given a Bayesian network, the global joint probability distribution over the set $V = \{A_1, ..., A_N\}$ can be expressed as a product of the N initial conditional probabilities via the following probabilistic *chain rule*:

$$p_c(A_1, ..., A_N) = \prod_{i=1..N} P(A_i \mid U_{A_i}). \tag{4.1}$$

The subscript c in p_c is used to mark that the joint distribution is computed via the chain rule.

Example 4.3 *Given the Bayesian network represented by the DAG of Figure 4.5 and the a priori and conditional probabilities given in Tables 4.1 and 4.2. The joint probability distribution is defined by:* $p_c(a \wedge b \wedge c) = P(c \mid a \wedge b) \cdot P(a) \cdot P(b), \forall abc$. *For instance,* $P_c(a_1 \wedge b_1 \wedge c_1) = 0.99 \cdot 0.01 \cdot 0.001 = 0.0000099$.

Table 4.1: A priori probabilities

a	$P(a)$	b	$P(b)$
a_1	0.01	b_1	0.001
a_2	0.99	b_2	0.999

Table 4.2: Conditional probabilities

c	a	b	$P(c \mid a \wedge b)$	c	a	b	$P(c \mid a \wedge b)$
c_1	a_1	b_1	0.99	c_2	a_1	b_1	0.01
c_1	a_1	b_2	0.9	c_2	a_1	b_2	0.1
c_1	a_2	b_1	0.5	c_2	a_2	b_1	0.5
c_1	a_2	b_2	0.1	c_2	a_2	b_2	0.99

Figure 4.5: Example of a DAG ($V = \{A, B, C\}$)

4.4.2 Recovering independence relations

In this Section, we present an important property of the probabilistic chain rule concerning the recovering of the independence relations implied by the DAG structure. We first present general definitions given by Pearl [103].

Definition 4.4 *A joint distribution M defined on V is said to be a* ***dependency model*** *if it is equipped with an independence relation I_M allowing to test for any three subsets of variables (X, Z, Y) of V the validity of the assertion "X is independent of Y given Z".*

It is clear that any probability distribution is a dependency model since we can test for any three subsets of variables (X, Z, Y) the validity of the assertion "X is independent of Y given Z" using the probabilistic independence relation I_{Prob} presented in Section 1.3.1.

Definition 4.5 *Let $\mathcal{G}$ be a DAG and let M be a dependency model. The DAG $\mathcal{G}$ is said to be an **I-map** (Independency map) of M if all the independence relations implied by its structure are valid in M. More formally:*

$$< X|Z|Y >_{\mathcal{G}} \Rightarrow I_M(X, Z, Y). \tag{4.2}$$

We will see later that this property is important especially when developing propagation algorithms since we will use the d-separation relations implied by the DAG structure $\mathcal{G}$ to simplify different calculus which is impossible if these relations are not valid in the joint distribution M. To test I-mapness, for any couple $(\mathcal{G}, M)$, Pearl proposes the following fundamental theorem [103]:

Theorem 4.1 *Let M be any dependency model such that I_M is a **semi-graphoid**[1] and let $\mathcal{G}$ be a DAG. If each variable $A \in V$, is conditionally independent of its non-descendants (Z_A) given its parents (U_A), then $\mathcal{G}$ is a minimal I-map of M.*

The probabilistic independence relation, I_{Prob}, is a semi-graphoid [103]. Moreover, in any joint distribution p_c computed using the chain rule (4.1), each variable is conditionally independent of *its non-descendants (Z_A) given its parents (U_A)* [103]. In other terms, any independence relation implied by the DAG structure can be recovered from the joint distribution relative to the Bayesian network and computed via the probabilistic chain rule (4.1).

4.5 Propagation in Bayesian networks

Given a joint probability distribution on the variable set $V = \{A_1, ..., A_N\}$, we can determine how the realization of specific values of some variables affects the probabilities of the remaining variables by marginalization. Unfortunately, assuming that we handle binary variables, we would need to compute 2^n probabilities, which is not realistic, even for small values of n.

Bayesian Networks simplify this problem by taking advantage of existing causal connections between nodes. Indeed, instead of computing the whole joint distribution relative to a Bayesian network, in order to compute for any variable of interest A_i the probability $P(A_i \mid e)$ (where e is the total evidence), it is possible to perform **local** computations using *probabilistic*

[1] *i.e. satisfies symmetry, decomposition, weak union and contraction (see Chapter 3).*

inference or propagation algorithms.

The task of calculating posterior marginals on nodes in an arbitrary Bayesian network is known to be **NP-complete** [30] except for singly connected graphs. Many algorithms where developed to perform propagation in Bayesian networks [32, 34, 84, 93, 94, 100, 103, 105] but we can partition them into two classes:

- **Exact algorithms**: The fundamental algorithm for exact probability propagation is the one proposed by Kim and Pearl [87] and Pearl [102, 103]. In this algorithm, the impact of each new piece of evidence is viewed as a perturbation that propagates in parallel through the network, via a message passing mechanism, between neighboring variables. In the case of singly connected networks this algorithm converges to the correct marginals in a number of iterations equal to the diameter[2] of the graph. A *centralized* version of this algorithm was proposed by Poet and Shachter [105] and converges in two iterations. However, as Pearl pointed [103], the same algorithm will not give the correct beliefs for multiply connected networks i.e. when loops are present. Approximate methods derived from this algorithm such as *loopy belief propagation* [100] where proposed for multiply connected networks.

  There are also a number of related propagation algorithms that operate on undirected graphs, namely the Lauritzen and Spiegelhalter algorithm [94]. The principle of this algorithm is to transform the initial graph into a junction tree which is a tree of clusters of variables. The messages will be transmitted between clusters allowing the computation of marginal distributions in two passes. A refinement of this algorithm leads to HUGIN propagation proposed by Jensen and al. [84].

- **Approximate algorithms**: Many models of interest, such as those with repetitive structure, as in multivariate time-series or image analysis, make exact inference very slow. In such cases, we should resort to approximation techniques which gives results approaching the exact solutions. Unfortunately, approximate inference is NP-complete [32], but we can nonetheless come up with approximations which often work well in practice. As examples of approximate methods we quote, *sampling (Monte Carlo) methods* [29], *loopy belief propagation* [100], which entails applying Pearl's algorithm to the original graph, even if it has loops, *bounded cutset conditioning* based on instantiating subsets of the variables, in order to break loops in the graph etc.

[2] a **diameter** of a graph is the maximum length of shortest paths between two vertices in the graph. For most regular graphs, it is a function of the number of nodes in the graph, but in general its value has to be computed.

In what follows, we focus on the most used exact propagation algorithms in singly and multiply connected DAGs.

4.6 Propagation in polytrees

In this Section, we briefly present the *centralized* version of Pearl's algorithm [87, 103] for probabilistic propagation in **polytrees** proposed in [105]. In this algorithm the impact of each new piece of evidence is viewed as a perturbation that propagates in parallel through the network via a message passing between neighboring variables. Local communication between variables is based on two kinds of messages, called λ-messages and μ-messages[3] circulating, respectively, from children to parents and from parents to children.

4.6.1 Notations

Let E be the subset of instanciated variables and e be its corresponding instance. The instance e corresponds to the total evidence in the graph. Considering any node A in the graph, the set E can be partitioned into the instanciated variables upstream of A denoted by E_A^+ and all the rest denoted by E_A^- i.e. the instanciated variables below A and A itself if it is instanciated (see Figure 4.6). These two sets are defined by:

- $E_A^- = \{E_A, E_{AY_1}^-, ..., E_{AY_m}^-\}$,
  where E_A corresponds to the node A and $E_{AY_j}^-$ the instanciated variables below the arc from A to Y_j,
- $E_A^+ = E/E_A^- = \{E_{U_1A}^+, ..., E_{U_nA}^+\}$,
  where $E_{U_iA}^+$ denotes the instanciated variables above the arc from U_i to A.

e_A^+ and e_A^- denote , respectively, the evidence attached to the nodes in E_A^+ and E_A^-. e_A denotes the evidence attached to the node A i.e.

$$e_A = \begin{cases} a & \text{if A is instanciated to a } (e_A = a) \text{ or A is not instanciated} \\ \emptyset & \text{otherwise } (\text{A is instanciated to } e_A \neq a) \end{cases}$$

In the same manner, $e_{AY_j}^-$ denotes the evidence relative to $E_{AY_j}^-$ and more precisely, the evidence below the edge from A to Y_j. Moreover, $e_{U_iA}^+$ denotes the evidence relative to $E_{U_iA}^+$ and more precisely, to the evidence above the arc from U_i to A.

Example 4.4 *Let us consider the DAG represented in Figure 4.2 and suppose that the instanciated variables are as follows:* $D = d_1, F = f_2, A = a_3, I = i_2, J = j_2, M = m_1$.

[3] we use μ instead of π, usually used in the literature, to avoid confusion with possibility distribution notation.

Figure 4.6: Partitioning of the evidence E with respect to the node A

Then, $E = \{A, D, F, I, J, M\}$, $E_A^+ = \{E_{BA}^-, E_{CA}^-\} = \{D, F\}$, $E_A^- = \{E_A, E_{AG}^-, E_{AH}^-\} = \{A, J, I, M\}$, $e = \{a_3, d_1, f_2, i_2, j_2, m_1\}$, $e_A^+ = \{e_{BA}^-, e_{CA}^-\} = \{d_1, f_2\}$, $e_A^- = \{e_A, e_{AG}^-, e_{AH}^-\} = \{a_3, j_2, i_2, m_1\}$.

4.6.2 Summary of propagation messages

The following summarizes the different definitions of messages[4] and values relative to a particular node $A \in V$. Each node is characterized by its conditional probability measure based on the total evidence, denoted by $Bel(A)$. This value depends on the λ-*value* and μ-*value* relative to the node A (denoted, respectively, by $\lambda(A)$ and $\mu(A)$).

The values of $\lambda(A)$ and $\mu(A)$ are computed from the λ-*messages* received from the children of A and μ-*messages* received from its parents (denoted, respectively, by $\lambda_{Y_j}(A)$ and $\mu_A(U_i)$ where $\lambda_{Y_j}(A)$ is the message that A receives from its child Y_j and $\mu_A(U_i)$ is the message that A receives from its parent U_i).

We now give the different expressions which will be used later in the propagation algorithm.

- $\forall a \in D_A$, the current conditional probability measure of a based on the total evidence e is defined by:

$$Bel(a) = P_c(a \mid e) = \alpha \cdot \lambda(a) \cdot \mu(a), \tag{4.3}$$

  where $\alpha = \frac{1}{\sum_a Bel(a)}$ is a normalization factor.

- The λ value relative to each instance $a \in D_A$ is defined in the following way:

$$\lambda(a) = P_c(a \mid e_A^-) = \lambda_A(a) \cdot \prod_{j=1}^{m} \lambda_{Y_j}(a), \tag{4.4}$$

  where $\lambda_A(a)$ denotes local evidence related to the node A such that:

$$\lambda_A(a) = \begin{cases} 0 & \text{if } e_A \neq a \text{ (A is instanciated to } (e_A \neq a) \\ 1 & \text{otherwise (A is instanciated to a } (e_A = a) \\ & \text{or A is not instanciated).} \end{cases}$$

[4] Proofs of these expressions are given in [103].

This means that the local evidence is stored within each node and that it is integrated in the λ value which is used for computing the λ-messages that each node sends to its parents.

- The μ value relative to each instance $a \in D_A$ is defined in the following way:

$$\mu(a) = P_c(a \mid e_A^+) = \sum_u P(a \mid u) \cdot \prod_{i=1}^{n} \mu_A(u_i). \tag{4.5}$$

- The λ messages from A to its parent $U_i, (i \in \{1, ..n\})$ when $U_i = u_i$ is defined by:

$$\lambda_A(u_i) = P_c(e^-_{U_i A} \mid u_i) = \beta \sum_a \lambda(a)[\sum_{u_k : k \neq i} P(a \mid u) \cdot \prod_{k \neq i} \mu_A(u_k)], \tag{4.6}$$

where β is a normalization constant.

If the graph is a rooted tree (each node has a unique parent), then this message is simplifies to:

$$\lambda_A(u_i) = \beta \sum_a \lambda(a) \cdot P(a \mid u).$$

- The μ messages from A to its child $Y_j, (j \in \{1, ..m\})$ when $A = a$ is defined by:

$$\mu_{Y_j}(a) = P_c(a \mid e^+_{AY_j}) = \alpha \cdot \lambda_A(a) \cdot \prod_{i=1..m, i \neq j} \lambda_{Y_i}(a) \cdot \mu(a). \tag{4.7}$$

These formulae are only slightly modified from those in Pearl [103] since local evidence $\lambda_A(A)$ (lambda from self) is stored within each node instead of creating *dummy* or *evidence* and must be included in the formulae for each node. Note that, contrary to [105], all these messages are normalized since the μ message is defined by $P_c(a \mid e_A^+)$ instead of $P_c(a \wedge e_A^+)$.

4.6.3 Propagation algorithm

Poet and Shachter [105] have proposed a revised polytree algorithm (centralized version) developed in a manner similar to the propagation algorithms in undirected graphs. Indeed, in the polytree algorithm proposed by Pearl [103] each node is visited at most one for each piece of evidence. The revised version decreases the number of messages since each node is visited at most twice no matter how much evidence is observed. Since the number of arcs in a singly connected graph is less than the number of nodes, this means that *the number of messages used in the revised algorithm is less than twice the number of nodes.* Let s be the size of the largest conditional probability table and N be the number of variables, then the theoretical complexity is $O(s * N)$.

This makes it particularly easy to implement. The main steps of this algorithm are as follows:

- *Choosing the root of propagation.* In this step, we should choose an arbitrary node within the smallest connected set of nodes containing the observed nodes, denoted by $\mathcal{S}$.

- *Initialization.* In this step, all λ and μ values and messages are initialized to the degree 1. For each root A, $\forall a \in D_A$ the value $\mu(a)$ is initialized to $P(a)$. For each observed node A, if A is instanciated to a, then $\lambda_A(a)$ is initialized to the degree 1, otherwise it is initialized to the degree 0.

- *Collect-evidence.* During, this phase each node in $\mathcal{S}$ first computes its λ and μ values using (4.4) and (4.5), then it passes a message to its adjacent node in the pivot direction. If this node is a parent, then the message is computed using (4.6), otherwise (i.e. it is a child) the message is computed using (4.7). The collect evidence starts with the node farthest from the pivot in $\mathcal{S}$.

- *Distribute-evidence.* During, this phase each node first computes its λ and μ values using (4.4) and (4.5), then it passes messages to its adjacent nodes away from the pivot direction, beginning with the pivot itself until reaching the leaves. The computation to the messages to sent to adjacent nodes depends on their type. Indeed, if the treated node is a parent, then the message is computed using (4.6), otherwise (i.e. it is a child) the message is computed using (4.7).

- *Marginalization.* For each node A, compute $Bel(A) = P_c(a \mid e)$ using (4.3).

Example 4.5 *Let us consider the DAG given in Figure 4.7(a) and suppose that we receive a certain information about E i.e. $E = e_3$ and about A i.e. $A = a_1$, then the observed nodes are E and A which means that $\mathcal{S} = \{A, B, C, D, E\}$. Suppose that the pivot node is E, then the message passing is summarized by Figure 4.7(b).*

Figure 4.7: Example of propagation on a singly connected DAG

4.7 Propagation in multiply connected DAGs

The algorithm presented in the previous Section is only adapted for polytrees. Indeed, as Pearl noted [103], the same algorithm will not give the correct beliefs for multiply connected networks. Thus, in this Section we present a more general established method for exact inference in Bayesian networks (with singly or multiply DAGs) known as *the probability propagation in*

junction trees. This method was developed by Lauritzen and Spiegelhalter [94] and refined by Jensen [84] and it is the basis of HIGIN propagation software. The principle of this propagation method is to process on a new graphical representation derived from the initial Bayesian network called **junction tree** and denoted by $\mathcal{JT}$ [84].

Each node in $\mathcal{JT}$ is a set of variables called *clusters* and denoted by C_i and each edge is labeled with the intersection of the adjacent clusters C_i and C_j called *separator* and denoted by S_{ij}; c_i and s_{ij} denote, respectively, the possible instances of the cluster C_i and the separator S_{ij}; $c_i[A]$ denotes the instance in c_i of the variable A. For each cluster C_i (resp. separator S_{ij}) of $\mathcal{JT}$, we assign a local joint distribution relative to the variables in the cluster (resp. separator), called *potential*. and denoted by ψ_{C_i} (resp. $\psi_{S_{ij}}$).

From $\mathcal{JT}$, we can associate a unique global joint probability distribution denoted by $p_{\mathcal{JT}}$ and defined by:

Definition 4.6 *The joint distribution associated with $\mathcal{JT}$ is expressed by:*

$$p_{\mathcal{JT}}(A_1, .., A_N) = \frac{\prod_{i=1}^{m} \psi_{C_i}}{\prod_{j=1}^{m-1} \psi_{S_{ij}}}, \tag{4.8}$$

where m is the number of clusters in $\mathcal{JT}$.

Definition 4.7 *Let C_i and C_j be two adjacent clusters in a junction tree $\mathcal{JT}$ and let S_{ij} be their separator. Then, the link between C_i and C_j is said to be **stable** or **consistent** if:*

$$\sum_{C_i \setminus S_{ij}} \psi_{C_i} = \psi_{S_{ij}} = \sum_{C_j \setminus S_{ij}} \psi_{C_j}. \tag{4.9}$$

*If all links in a junction tree are consistent, then the junction tree is said to be **globally consistent**.*

Jensen has shown that if a junction tree is globally consistent, then the potential of each cluster C_i satisfies:

$$\psi_{C_i} = P_c(C_i). \tag{4.10}$$

Using this fact, we can compute the probability distribution of any variable A, using any cluster C_i containing it by marginalizing ψ_{C_i} on A as follows:

$$P_c(A) = \sum_{C_i \setminus A} \psi_{C_i}. \tag{4.11}$$

Figure 4.8: Case of multiply connected DAG

4.7.1 Building junction trees

Given a Bayesian network with a DAG $\mathcal{G}$, it is possible to construct many junction trees corresponding to $\mathcal{G}$. However, it is important to note that the computational time required for propagation depends on clusters size in the junction tree.

The building procedure depends on the DAG structure. Indeed, if the DAG is singly connected then it is easy to construct its junction tree by forming for each node (except roots) a cluster containing it with its parents. Then, between any two clusters with a non-empty intersection, we should add a link with the intersection as a separator. If the resulting graph contains cycles, then they can be broken by removing any of their links since all separators on the cycle contain the same variable.

In the case of multiply connected graphs the situation can be more complicated as illustrated by the following example.

Example 4.6 *Let us consider the DAG of Figure 4.8 (a), then if we apply the procedure described for singly connected DAGs, we obtain the graph of Figure 4.8 (b) and we can see that the cycle cannot be broken.*

Before presenting the general procedure of building junction tree, we give further definitions:

Definition 4.8 *Given a DAG $\mathcal{G}$, the* ***moral graph*** *corresponding to $\mathcal{G}$, denoted by $\mathcal{MG}$, is obtained by* marrying parents *of each node in $\mathcal{G}$ and by dropping the direction of edges.*

Definition 4.9 *A moral graph $\mathcal{MG}$ is said to be* ***triangulated*** *if and only if every cycle of length four or greater contains an edge that connects two non-adjacent nodes in the cycle.*

The general procedure that allows the construction of a junction tree for any DAG structure can be performed via the following three steps [82]:
- Moralization of the initial DAG $\mathcal{G}$,
- Triangulation of the moral graph,
- Building a junction tree from the triangulated moral graph.

We will see that steps 2 and 3 are *non-deterministic* which explains the fact that different junction trees can be built from the same DAG.

Step 1: Constructing the moral graph. Given a DAG $\mathcal{G}$, the moral graph $\mathcal{MG}$ corresponding to $\mathcal{G}$ is obtained through the following procedure:

Algorithm 4.1: Constructing the moral graph

begin

- Create the undirected graph $\mathcal{UG}$ from $\mathcal{G}$ by dropping the directions of the arcs;
- Create $\mathcal{MG}$ from $\mathcal{UG}$ by connecting the parent set of each nodes (by adding edges to $\mathcal{UG}$);

end

Example 4.7 *Figure 4.10 represents the moral graph corresponding to the DAG of Figure 4.9. The edges added to $\mathcal{G}$ are shown as dashed lines (i.e. the arc between D and E parents of F).*

Figure 4.9: A Direct Acyclic Graph

Figure 4.10: Moral Graph of the DAG in Figure 4.9

As shown in Example 4.6, the moralization can induce unbroken cycles, that is why we should *triangulate* the moral graph in order to avoid them.

Step 2: Triangulating the moral graph and identifying clusters. It is possible to have different triangulations of a moral graph. In particular we can simply construct a unique cluster containing all the variables. However such triangulation is not interesting since it does not allow local computations.
The task of finding an optimal triangulation is stated as an *NP-complete* problem [30] and several heuristics where proposed. We now present a triangulation procedure using a node selection criterion which is a greedy polynomial-time heuristic that produces high-quality triangulations in real world settings [88]. The following procedure also identifies the cluster set (denoted by *Cluster_set*) using Golumbic algorithm [73] which ensures that this set should not contain any cluster properly contained in a largest one.

Figure 4.11: Triangulation of the moral graph in Figure 4.10

Algorithm 4.2: Triangulating the moral graph and identifying clusters

begin
- Cluster_set $\leftarrow \emptyset$;
- $\mathcal{MG}' \leftarrow \mathcal{MG}$;
- **while** *there are still nodes left in* $\mathcal{MG}'$ **do**
  - **if** *there exists a node* A *such that all its adjacent nodes are connected* **then**
    - Form a cluster C containing A and its adjacent nodes;
    - Delete A from $\mathcal{MG}'$;
  - **else**
    - Find a node A with the smallest number of adjacent nodes;
    - Add edges to connect all of these adjacent nodes;
    - For each added edge to $\mathcal{MG}'$, add the same corresponding edge to $\mathcal{MG}$;
    - Form a cluster C containing A and its adjacent nodes;
    - Delete A from $\mathcal{MG}'$;
  - **if** $C \not\subset Cluster\_set$ **then** $Cluster\_set \leftarrow Cluster\_set \cup \{C\}$;

end

Example 4.8 *Figure 4.11 represents the triangulated graph, as obtained from the moral graph in Figure 4.10. The dashed lines indicate the edges added to triangulate the moral graph. The eliminating ordering is represented in Table 4.3.*

Table 4.3: Eliminating ordering

Eliminated node	*Induced cluster*	*Added edges*	*Cluster_set*
F	DEF	*none*	$\{DEF\}$
C	ACE	*(A, E)*	$\{DEF, ACE\}$
B	ABD	*(A, D)*	$\{DEF, ACE, ABD\}$
D	DAE	*none*	$\{DEF, ACE, ABD, DAE\}$
E	EA	*none*	$\{DEF, ACE, ABD, DAE\}$ $(EA \subseteq ACE)$
A	A	*none*	$\{DEF, ACE, ABD, DAE\}$ $(A \subseteq ACE)$

Step 3: Building an optimal junction tree. To build an *optimal* junction tree, we should connect the clusters identified in the previous step such that all clusters on the path between any two clusters C_i and C_j should contain $C_i \cap C_j$. The *optimality* criterion is useful to minimize the computational time required for propagation. Indeed, if the triangulation

generates clusters with a high weight, then the computations will be very complex and even impossible to perform.

Given a set of m clusters (i.e. *Cluster_ set*), we can build a junction tree by iteratively inserting separators between pairs of clusters (denoted by *Separator_ set*) as follows:

Algorithm 4.3: Building an optimal junction tree

```
begin
    for i ← 1 to m do
        Separator_set ← ∅;
        C_i ← Cluster_set[i];
        for j ← i to (m-1) do
            C_j ← Cluster_set[j];
            Create a candidate separator S_ij for C_i and C_j;
            Separator_set ← S_ij;
            Select a separator S_ij from Separator_set according to the criterion
            specified in the sequel;
        Insert the separator S_ij between the clusters C_i and C_j;
end
```

Choosing the appropriate separators: In order to describe the criterion for selecting separators (in the algorithm above), we need the following definitions:
- The *weight* of a node A is the number of its values.
- The *weight* of a cluster C_i is the product of the weights of its variables.
- The *mass* of a separator S_{ij} is the number of variables it contains, or the number of variables in $C_i \cap C_j$.
The *cost* of a separator S_{ij} is the product of the weights of its variables.

To ensure that the resulting junction tree satisfies the junction tree property, we should choose the candidate separator with the *largest mass*. When two or more separators of equal mass can be chosen, we can optimize the inference time on the resulting junction tree by choosing the candidate separator with the *smallest cost* [85].

Example 4.9 *Let us continue Example 4.8 and suppose that the variables are binary variables. Thus, using Cluster_ set=$\{ABD, ACE, ADE, DEE\}$ identified above, we obtain the values in Table 4.4. Then, we will choose the connecting separators DE, AE and AD to built the junction tree illustrated in Figure 4.12.*

In the following, we denote by $\psi^t_{C_i}$ the potential of the cluster C_i at a step t of the

Table 4.4: Choosing the appropriate separators

Clusters	*Candidate separator*	*Mass*	*Cost*	*Choosen separator*
DEF, ACE	*E*	*1*	*2*	
DEF, ABD	$\emptyset$	-	-	
DEF, ADE	*DE*	*2*	*4*	*
ACE, ABD	*A*	*1*	*2*	
ACE, ADE	*AE*	*2*	*4*	*
ABD, ADE	*AD*	*2*	*4*	*

Figure 4.12: Junction tree of the DAG in Figure 4.9

propagation. $t = I$ (resp. $t = C$) corresponds to the initialization (resp. global consistency) step.

4.7.2 Propagation algorithm

Given the transformation of the DAG into a junction tree, the main steps of probability propagation in junction trees with no evidence are as follows:

Step 1: Initialization. The first step in the propagation is to initialize the junction tree using the initial conditional distributions:

Algorithm 4.4: Initialization

begin

For each cluster C_i: $\psi^I_{C_i} \leftarrow 1$;

For each separator S_{ij}: $\psi^I_{S_{ij}} \leftarrow 1$;

For each variable A, choose a cluster C_i containing $\{A\} \cup U_A$: $\psi^I_{C_i} \leftarrow \psi^I_{C_i} \cdot P(A \mid U_A)$;

end

The resulted potentials encode the joint distribution relative to the junction tree. Namely,

$$p_c = p^I_{\mathcal{JT}}, \tag{4.12}$$

where p_c is the joint distribution encoded by the Bayesian network (using 4.1) and $p^I_{\mathcal{JT}}$ is the joint distribution encoded by $\mathcal{JT}$ (using (4.8)).

The initialized junction tree can be *inconsistent* since this initial assignment does not guarantee the global consistency requirement of Equation (4.9). Thus, we should run the the second step of global propagation which ensures global consistency.

Step 2: Global propagation. The global propagation is performed by a *message passing* mechanism between clusters until reaching the global consistency of the junction tree. Given a junction tree with m clusters, the global propagation algorithm begins by choosing an arbitrary cluster to be the *pivot node* and then performing $2*(m-1)$ messages passes, divided into two phases:

- A *collect-evidence* phase in which each cluster passes a message to its adjacent cluster in the pivot direction, beginning with the clusters farthest from the pivot (which correspond to leaves). The order in which messages are sent is denoted by the *postorder* of the pivot.
- A *distribute-evidence* phase in which each cluster passes messages to its adjacent clusters away from the pivot direction, beginning with the pivot itself. In this phase messages circulate from the pivot until the leaves are reached. The order in which messages are sent is denoted by the *preorder* of the pivot.

Example 4.10 *Suppose that in the Bayesian network given in Example 4.6, we receive a certain information on C and E. Then, we can choose the cluster ACE as the pivot, in this case* Postorder $= [ABD, DEF, ADE, ACE]$ *and* Preorder $= [ACE, ADE, DEF, ABD]$ *and the message passing is represented by Figure 4.13.*

Figure 4.13: Message passes during global propagation

In both phases of collect and distribute evidence, if a cluster C_i sends a message to its adjacent cluster C_j, then the potentials of C_i, C_j and their separator S_{ij} are updated as follows:

1. Save the same potential for C_i
$$\psi_{C_i}^{t+1} \leftarrow \psi_{C_i}^{t}. \tag{4.13}$$

2. Assign a new potential to S_{ij}
$$\psi_{S_{ij}}^{t+1} \leftarrow \sum_{C_i \setminus S_{ij}} \psi_{C_i}^{t}. \tag{4.14}$$

3. Assign a new potential to C_j:
$$\psi_{C_j}^{t+1} \leftarrow \psi_{C_j}^{t} \cdot \frac{\psi_{S_{ij}}^{t+1}}{\psi_{S_{ij}}^{t}}. \tag{4.15}$$

The outline of the global propagation procedure is as follows:

Algorithm 4.5: Global propagation

begin
  Choosing the root of propagation
  - Let *Pivot* be an arbitrary cluster C_i to represent the root of propagation;
  - Let *Postorder* be the vector containing the order in which messages are sent in the distribute-evidence phase (the last node is the pivot);
  - Let *Preorder* be the vector containing the order in which messages are sent in the collect-evidence phase (the first node is the pivot);
  Collect-evidence
  for *i ← 1 to length(Postorder)-1* **do**
    $C_i \leftarrow$ *Postorder*[i];
    $C_j \leftarrow$ adjacent cluster of C_i in *Postorder*;
    Post a message from C_i to C_j using (4.13), (4.14) and (4.15);
  Distribute-evidence
  for *i ← 1 to length(*Preorder*)* **do**
    $C_i \leftarrow$ *Preorder*[i];
    Below ← adjacent clusters of C_i in *Preorder*;
    for *j ← 1 to length(Below)* **do**
      $C_j \leftarrow$ Below[j];
      Post a message from C_i to C_j using (4.13), (4.14) and (4.15);
end

In this message passing algorithm, a cluster passes a message to an adjacent cluster only after it has received messages from all its other adjacent clusters. This condition ensures global consistency of the junction tree when global propagation is completed [84].

Step 3: Marginalization. Using the consistent junction tree obtained from the previous phase, we can now compute for each variable of interest A, the probability $P_c(A)$ as follows:

Algorithm 4.6: Marginalization

begin

Identify a cluster C_i containing A;

Compute $P_c(A)$ by marginalization of $\psi^C_{C_i}$ on A: $P_c(A) \leftarrow \sum_{C_i \setminus A} \psi^C_{C_i}$;

end

Handling the evidence. When treating the more general problem of computing $P_c(A \mid e)$, where e is the total evidence, the initialization procedure will be extended by these two additional steps:

- For any instanciated variable A, encode the observation $A = a$ as a *likelihood* Λ_A defined by:

$$\Lambda_A(a) = \begin{cases} 1 & \text{if } A \text{ is not instanciated} \\ 1 & \text{if } A \text{ is instanciated for } a \\ 0 & \text{if } A \text{ is instanciated but not for } a. \end{cases} \tag{4.16}$$

- Identify a cluster C_i containing A: $\psi^I_{C_i} \leftarrow min(\psi^I_{C_i}, \Lambda_A)$.

By entering the observation set, the junction tree encodes $p_{\mathcal{JT}}(V \wedge e)$ instead of $p_{\mathcal{JT}}(V)$. Then, when we marginalize any cluster potential $\psi^C_{C_i}$ into a variable A (s.t. $A \subseteq C_i$) (see (4.11), we obtain the probability of A and e:

$$P_c(A \wedge e) = \sum_{C_i \setminus A} \psi^C_{C_i}.$$

However, our goal is to compute the conditional probability $P_c(A \mid e)$, this value can be easily obtained from $P_c(A \wedge e)$ as follows:

$$P_c(A \mid e) = \frac{P_c(A \wedge e)}{P_c(e)} = \frac{P_c(A \wedge e)}{\sum_A P_c(A \wedge e)}. \tag{4.17}$$

4.8 Valuation Based Systems

In this Section, we give a brief presentation of Valuation Based Systems (VBS) which are more general graphical models proposed by Shenoy [111, 112, 113, 115, 114] allowing the

representation of both decision and uncertain problems. The Framework of Valuation Based Systems [116] is sufficiently abstract to include different formalisms as probability theory, belief functions, kappa functions and possibility theory using only the product operator [112]. . A Valuation Based System over a set of variables V consists of two components:

- a *graphical component* called a *valuation network*. In a valuation network:
  - *circular* nodes represent random variables,
  - *rectangular* nodes represent decision variables,
  - *triangular* nodes represent potentials,
  - *diamond-shaped* nodes represent utility functions,
  - *edges* (i.e. undirected lines) linking variables to potentials and utility functions denote the domains of these functions,
  - *arcs* (i.e. directed lines) between variables define the information constraints in the sense that an arc from a random variable A to a decision variable B means that the true value of A is known at the time an act from D_A has to be chosen and an arc from a decision variables B to a random variable A means that the true value of A is only revealed after an act from D_A has been chosen.
- a *numerical component* representing our knowledge by functions called **valuations**. Valuations are functions that assign values to the elements or frames for sets of variables and they can be combined or marginalized. Valuations can be interpreted as probabilities, possibilities, belief functions etc.

4.8.1 Combination and marginalization

Let $\mathcal{V}$ denote the set of all valuations. The inference in VBS uses the two basic operations of **combination** and **marginalization** defined as follows:

- A *combination* is a mapping $\otimes : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{V}$, such that
  1. if ρ is a valuation on X and σ is a valuation on Y, then $\rho \otimes \sigma$ is a valuation of $X \cup Y$,
  2. if either ρ or σ is not a nonzero valuation, then $\rho \otimes \sigma$ is not a nonzero valuation,
  3. if ρ and σ are both nonzero valuation, then $\rho \otimes \sigma$ may or may not be a nonzero valuation.
- A *marginalization* is a mapping $\downarrow (X \setminus \{A\}) : \mathcal{V}_X \rightarrow \mathcal{V}_{X \setminus \{A\}}$, such that
  1. if σ are valuations X, then $\sigma^{\downarrow (X \setminus \{A\})}$ is a valuation for $(X \setminus \{A\})$,
  2. $\sigma^{\downarrow (X \setminus \{A\})}$ is a nonzero valuation iff σ is a nonzero valuation.

4.8.2 Inference in a VBS

In a VBS the combination of all valuations is called the *joint valuation*. Given a VBS the inference is made by computing the marginal of the joint valuation for each variable in the system. We first present three axioms that enable local computation of marginals of the joint valuation [116]:

- **A1**: *Commutativity and associativity of combination*
  Given three valuations ρ, σ, and τ relative, respectively, for X, Y, and Z then,

  $$\rho \otimes \sigma = \sigma \otimes \rho \text{ and } \rho \otimes (\sigma \otimes \tau) = (\rho \otimes \sigma) \otimes \tau).$$

  This axiom implies that the order in which the combination is made is not important.

- **A2**: *Order of deletion does not matter*
  Suppose σ is a valuation for X, and suppose $A, B \in X$, the:

  $$(\sigma^{\downarrow(X\setminus\{A\})})^{\downarrow(X\setminus\{A,B\})} = (\sigma^{\downarrow(X\setminus\{B\})})^{\downarrow(X\setminus\{A,B\})}.$$

  This axioms implies that the order in which the variables deleted is not important. However, different deletion sequences may involve different computational efforts. Note that the task of finding an optimal deletion sequence has been shown as an NP-complete problem [1]. But, there are several heuristics for finding good deletion sequences [88].

- **A3**: *Distributivity of marginalization over combination*
  Given two valuations ρ and σ relative, respectively, to X, Y, and Z and a variable $Anot \in Y$, and $A \in X$ then,

  $$(\rho \otimes \sigma)^{\downarrow((X\cup Y)\setminus\{A\})} = \rho \otimes (\sigma^{\downarrow(X\setminus\{A\})}).$$

  This axiom is crucial to make local computations since it implies that we can compute $(\rho \otimes \sigma)^{\downarrow((X\cup Y)-\{A\})}$ without computing $\rho \otimes \sigma$ The combination operation in $\rho \otimes \sigma$ is on the frame for $X \cup Y$ whereas the combination operation in $(\rho \otimes \sigma)^{\downarrow((X\cup Y)-\{A\})}$ is on the frame for $(X \cup Y) - \{A\}$.

Given a VBS, $\{\{\sigma_1, ..., \sigma_m\}, \otimes, \downarrow\}$ with n variables and m valuations. Suppose we have to compute the marginal of the joint valuation for a variable A, $(\sigma_1 \otimes ... \otimes \sigma_m)^{\downarrow\{A\}}$. The basic idea of the **fusion algorithm** is to successively delete all variables but A from the VBS. The variables maybe deleted in any sequence (see Axiom **A2**), but different deletion sequences may involve different computational costs. When we delete a variable, we have to do a *fusion*

operation on the valuations.

Let $\sigma_1, ..., \sigma_k$ be a set of k valuations, such that σ_i is a valuation for the subset of variables X_i. Let $Fus_X\{\sigma_1, ..., \sigma_k\}$ denote the collection of valuations after fusing the valuations in the set $\{\sigma_1, ..., \sigma_k\}$ w.r.t the variable A. Then,

$$Fus_X\{\sigma_1, ..., \sigma_k\} = \{\sigma^{\downarrow(X-\{A\})}\} \cup \{\sigma_i \mid A \notin X_i\}, \tag{4.18}$$

where $\sigma = \otimes\{\sigma_i \mid A \in X_i\}$ and $X = \bigcup\{X_i \mid A \in X_i\}$.

Given the sequence of deletion $A_1, A_2, ..., A_{n-1}$, Shenoy shows that

$$(\sigma_1 \otimes ... \otimes \sigma_m)^{\{A\}} = \otimes\{Fus_{A_{n-1}}\{...Fus_{A_2}\{Fus_{A_1}\{\sigma_1, ..., \sigma_m\}\}\}\}. \tag{4.19}$$

After fusion, the set of valuations is changed such that all valuations that bear on A are combined and the resulting valuation is marginalized. The other valuations (that do not bear on A) remain unchanged.

4.9 Conclusion

In this Chapter we have presented basic definitions related to probabilistic Bayesian networks. Moreover, we have detailed exact propagation algorithms in these networks. Namely, Pearl's propagation algorithm [103, 105] and propagation in junction trees [84]. Pearl's propagation is polynomial but only deals with singly connected DAGs while the junction tree algorithm is NP-complete but deals with any DAG structure.

Chapters 5 presents possibilistic counterparts of Bayesian networks. Then, Chapter 6 gives the adaptation of the centralized version of Pearl's propagation and propagation in junction trees, for product-based and min-based possibilistic networks.

Chapter 5

Basic Definitions of Possibilistic Networks

5.1 Introduction

As mentioned in Chapter 1, the possibilistic logic [52] has been well defined contrary to possibilistic networks. Indeed, only few works exist on directed causal possibilistic networks. Indeed, existing works on possibilistic graphical models are either a direct adaptation of probabilistic approaches (without any care to knowledge representation) or a way to perform learning from imprecise data [69, 24]. Regarding the possibilistic propagation, we can mention the possibilistic propagation in hypergraphs proposed by Dubois and Prade [55], the adaptation of Pearl's algorithm by Fonck [63]. Gebhardt, Kruse and Borgelt have proposed a software for possibilistic propagation in undirected possibilistic networks, called POSSINFER [22, 68, 69]. Shenoy [112] has also proposed a possibilistic version of the propagation algorithm in *Valuation Based Systems* using only the product operator. Cano, Delgado and Moral [28] have presented a propagation system in singly connected graphs encoded by *valuations*. A generalized version of this algorithm, to the case of multiply connected DAGs, has been proposed by Fonck [63].

The possibility theory offers two definitions of conditioning (see Section 1.4.3) one based on the minimum operator and the other based on the product operator. This leads to two possible definitions of directed causal possibilistic networks:
- *product-based possibilistic networks*: this kind of networks are very close to the probabilistic ones, as it will be explained later.
- *min-based possibilistic networks*: such networks have a different behaviour when comparing them to probabilistic ones. One major difference concerns the so called *coherence* problem

which is the non-recovering of initial data. We show that this problem should not be seen as a drawback especially on independence relations.

This Chapter presents these two kinds of networks. Section 5.2 defines directed causal possibilistic networks. Section 5.3 investigates product-based possibilistic networks and Section 7.2 deals with min-based possibilistic networks and treats, in particular, the coherence problem.
Proofs of this Chapter are given in Appendix C.
Results on coherence in min-based possibilistic networks are published in [9].

5.2 Definition of directed causal possibilistic networks

As in probabilistic Bayesian networks, a directed causal possibilistic network over a set of variables V is characterized by:
- A *graphical component* composed of a Directed Acyclic Graph (DAG) $\mathcal{G}$. The DAG structure encodes a set of independence relations exactly as in the probabilistic case. The choice of independence relation depends on the way used in combining local conditional possibilities.
- A *numerical component* consisting in a quantification of different links in the DAG using the conditional possibilities of each node in the context of its parents. Such conditional distributions should respect the following normalization constraints for each variable A:

- if $U_A = \emptyset$ (i.e. A is a root), then the a priori possibility relative to A should satisfy:

$$max_a \Pi(a) = 1, \forall a \in D_A,$$

- if $U_A \neq \emptyset$, then the conditional distribution of A in the context of its parents should satisfy:

$$max_a \Pi(a \mid u_A) = 1, \forall a \in D_A, u_A \in D_{U_A}.$$

If A is a binary variable, then $max(\Pi(a \mid u_A), \Pi(\neg a \mid u_A)) = 1$.

5.3 Product-based possibilistic networks

A *product-based possibilistic graph* over a set of variables V, denoted by ΠG_p, is a possibilistic graph where conditionals are defined using product-based conditioning (1.18), namely,

$$\pi(\omega \mid_p \phi) = \begin{cases} \frac{\pi(\omega)}{\Pi(\phi)} & \text{if } \omega \in \phi \\ 0 & otherwise. \end{cases}$$

Product-based possibilistic networks are appropriate for a numerical interpretation of the possibilistic scale.

The joint distribution relative to product-based possibilistic networks, denoted be π_p can be computed in the same manner that in Bayesian networks via the following **product-based** chain rule:

Definition 5.1 *(**Product-based chain rule**) Given a product-based possibilistic network ΠG_p, the global joint possibility distribution over the variable set $V = \{A_1, A_2, ..., A_N\}$ can be expressed as the product of the N initial a priori and conditional possibilities via the following product-based chain rule:*

$$\pi_p(A_1, ..., A_N) = \prod_{i=1..N} \Pi(A_i \mid U_{A_i}), \tag{5.1}$$

where $\prod$ is the product operator.

The product-based chain rule is obtained from the (product) independence relations induced by the DAG structure and local product-based conditional degrees.

Indeed, we recall that the Prod-independence relation, introduced in Section 3.3.1, is defined by $\Pi(x \mid y \wedge z) = \Pi(x \mid z), \forall x, y, z$ to express that the variable sets X and Y are Prod-independent in the context Z. We also recall that in the DAG "*each node is independent of its non-descendants in the context of its parents*".

Therefore, the product-based chain rule can be explained in the following way:
Let $d = (A_1, ..., A_N)$ be an ordering of the variables in V such that $\forall A_i, U_{A_i} \subseteq \{A_{i+1}, ..., A_N\}$.
Then, using product-based conditioning we have:
$\pi_p(A_1, ..., A_N) = \Pi(A_1 \mid A_2, ..., A_N) \cdot \Pi(A_2, ..., A_N)$
$= \Pi(A_1 \mid U_{A_1}) \cdot \Pi(A_2, ..., A_N)$ (since A_1 is independent of its non-descendants in the context of its parent set U_{A_1})
Iterating the same operation on the rest of variables leads to (5.1).

Example 5.1 *Let us consider the product-based possibilistic causal network presented by the DAG of Figure 5.1 and the a priori and conditional possibility distributions given in Table 5.1. Using the chain rule (5.1), we obtain the joint possibility distribution given in Table 5.2.*

The chain rule relative to any causal network should satisfy some important properties, especially concerning the *recovering of initial data* and of *independence relations* encoded by the DAG structure. In what follows, we study these two properties regarding the product-based independence relation.

Figure 5.1: DAG of example 5.1

Table 5.1: Initial distributions

b	$\Pi(b)$	c	$\Pi(c)$	a	b	c	$\Pi(a \mid b \wedge c)$	a	b	c	$\Pi(a \mid b \wedge c)$	d	a	$\Pi(d \mid a)$
b_1	*1*	c_1	*1*	a_1	b_1	c_1	*1*	a_2	b_1	c_1	*0.2*	d_1	a_1	*1*
b_2	*0.3*	c_2	*0.7*	a_1	b_1	c_2	*0.6*	a_2	b_1	c_2	*1*	d_1	a_2	*0.2*
				a_1	b_2	c_1	*1*	a_2	b_2	c_1	*0.1*	d_2	a_1	*0*
				a_1	b_2	c_2	*0.2*	a_2	b_2	c_2	*1*	d_2	a_2	*1*

5.3.1 Recovering initial data

The following proposition shows that, like the probabilistic case, the product-based chain rule allows the recovering of initial local distributions. In other terms, when we compute the joint possibility distribution using (5.1) we always recover the initial values provided by experts.

Proposition 5.1 *Let π_p be the global joint possibility distribution of ΠG_p computed using (5.1). Let $\Pi(a \mid u_A)$ be the conditional distribution given by the expert on the node A and $\Pi_p(a \mid u_A)$ be the conditional possibility computed from π_p. Then,*

$$\Pi_p(a \mid u_A) = \Pi(a \mid u_A). \tag{5.2}$$

The proof of this Proposition uses the following technical Lemma [16]:

Lemma 5.1 *Let X be a strict subset of V and x be a fixed instantiation of X. Let $Z = V/X$. Then,*

$$max_z\{\prod\{\Pi(a \mid u_A) : a \in z, u_A \subseteq z \wedge x\}\} = 1. \tag{5.3}$$

This Lemma means that the marginalization always provides normalized distributions. In particular, if we consider that the set X is empty (i.e. $Z = V$) then, the global joint distribution is normalized.

Table 5.2: Global joint distribution using product-based chain rule (5.1)

a	b	c	d	$\pi_p(a \wedge b \wedge c \wedge d)$	a	b	c	d	$\pi_p(a \wedge b \wedge c \wedge d)$
a_1	b_1	c_1	d_1	*1*	a_2	b_1	c_1	d_1	*0.04*
a_1	b_1	c_1	d_2	*0*	a_2	b_1	c_1	d_2	*0.2*
a_1	b_1	c_2	d_1	*0.42*	a_2	b_1	c_2	d_1	*0.14*
a_1	b_1	c_2	d_2	*0*	a_2	b_1	c_2	d_2	*0.7*
a_1	b_2	c_1	d_1	*0.3*	a_2	b_2	c_1	d_1	*0.006*
a_1	b_2	c_1	d_2	*0*	a_2	b_2	c_1	d_2	*0.03*
a_1	b_2	c_2	d_1	*0.042*	a_2	b_2	c_2	d_1	*0.042*
a_1	b_2	c_2	d_2	*0*	a_2	b_2	c_2	d_2	*0.21*

Example 5.2 *Let us consider the joint distribution π_p calculated in Example 5.1. We first illustrate Lemma 5.1. Let $X = \{A, B, C\}, x = a_1 \wedge b_2 \wedge c_2$, and $Z = \{D\}$, then we can check that $max(\Pi(d_1 \mid a_1), \Pi(d_2 \mid a_1)) = max(1, 0.2) = 1$.*
Using π_p, we can also check that $\Pi_p(a_1 \mid b_1 \wedge c_2) = \Pi(a_1 \mid b_1 \wedge c_2) = 0.6$, since $\Pi(a_1 \mid b_1 \wedge c_2) = \frac{\Pi_p(a_1 \wedge b_1 \wedge c_2)}{\Pi_p(b_1 \wedge c_2)} = \frac{0.42}{0.7} = 0.6$.

5.3.2 Recovering independence relations

As shown in the previous Chapter, the DAG structure encodes many independence relations via the d-separation relations, thus we should check that these relations can be recovered from the joint distribution computed using (5.1). This property is important especially when developing the propagation algorithms, as we will see in Chapter 6.

In other terms, we should check that for any three disjoint subsets of V, X, Y and Z, if X and Y are d-separated by Z, then X and Y are Prod-independent in the context Z (since we use the product independence as independence relation relative to ΠG_p).

As mentioned in Chapter 4, Pearl [103] states that to satisfy this condition the independence relation should be a semi-graphoid (i.e. satisfies symmetry, decomposition, weak union and contraction). Moreover, in the joint distribution computed via the chain rule, each variable $A \in V$ should be conditionally independent of *its non-descendants (Z_A) given its parents (U_A)*.

These two conditions are satisfied by the Prod-independence relation. Indeed, as mentioned in Chapter 3, this relation is a semi-graphoid [63], moreover it satisfies the following proposition

Proposition 5.2 *Let ΠG_p be a product-based possibilistic network. Let π_p be the joint possibility distribution computed using 5.1. Then, each variable $A \in V$, is Prod-independent of the variables in Z_A given its parent set U_A i.e. $\forall a \in D_A, \forall u_A \in D_{U_A}, \forall z_A \in D_{Z_A}$:*

$$\Pi_p(a \mid z_A \wedge u_A) = \Pi_p(a \mid u_A). \tag{5.4}$$

The proof of this proposition is similar to the one relative to the probabilistic independence relation [103] since the Prod-independence is expressed in the same manner than this relation.

Example 5.3 *Let us consider the node D in the ΠG_p given in Example 5.1. We can check, for instance, that D is Prod-independent of $Z_D = \{A, B, C\}$ in the context of $U_D = \{A\}$. Indeed, $\forall a, b, c, d, \Pi_p(d \mid a \wedge b \wedge c) = \Pi_p(d \mid a)$ as shown in Tables 5.3 and 5.4 derived from Table 5.2.*

Table 5.3: Conditional distribution of D in the context of A, B and C

a	b	c	d	$\Pi_p(d \mid a \wedge b \wedge c)$	a	b	c	d	$\Pi_p(d \mid a \wedge b \wedge c)$
a_1	b_1	c_1	d_1	*1*	a_2	b_1	c_1	d_1	*0*
a_1	b_1	c_1	d_2	*0.2*	a_2	b_1	c_1	d_2	*1*
a_1	b_1	c_2	d_1	*1*	a_2	b_1	c_2	d_1	*0*
a_1	b_1	c_2	d_2	*0.2*	a_2	b_1	c_2	d_2	*1*
a_1	b_2	c_1	d_1	*1*	a_2	b_2	c_1	d_1	*0*
a_1	b_2	c_1	d_2	*0.2*	a_2	b_2	c_1	d_2	*1*
a_1	b_2	c_2	d_1	*1*	a_2	b_2	c_2	d_1	*0*
a_1	b_2	c_2	d_2	*0.2*	a_2	b_2	c_2	d_2	*1*

Table 5.4: Conditional distribution of D in the context of A

d	a	$\Pi_p(d \mid a)$
d_1	a_1	*1*
d_1	a_2	*0*
d_2	a_1	*0.2*
d_2	a_2	*1*

Thus, using Proposition 5.2 and Corollary 4.1, we can give the following fundamental proposition:

Proposition 5.3 *Let ΠG_p be a product-based possibilistic network. Let π_p be the joint distribution relative to ΠG_p. Let X, Y and Z be three disjoint subsets of V. If X and Y are d-separated by Z in G, then X and Y are Prod-independent in the context of Z in π_p. More formally:*

$$< X|Z|Y >_{\mathcal{G}} \Rightarrow I_{Prod}(X, Z, Y). \tag{5.5}$$

In other terms, any independence relation implied by the DAG structure of a product-based possibilistic networks ΠG_p, can be recovered from the joint distribution relative to ΠG_p computed via the product-based chain rule (5.1).

5.4 Min-based possibilistic networks

A *min-based possibilistic graph* over a set of variables V, denoted by ΠG_m, is a possibilistic graph where conditionals are defined using min-based conditioning (1.17), namely,

$$\pi(\omega \mid_m \phi) = \begin{cases} 1 & if\ \pi(\omega) = \Pi(\phi) \text{ and } \omega \in \phi \\ \pi(\omega) & if\ \pi(\omega) < \Pi(\phi) \text{ and } \omega \in \phi \\ 0 & otherwise. \end{cases}$$

min-based possibilistic networks are appropriate for an ordinal interpretation of the possibilistic scale.

The joint distribution relative to min-based possibilistic networks, denoted by π_m can be computed via the following **min-based** chain rule:

Definition 5.2 *(**min-based chain rule**) Given a min-based possibilistic network ΠG_m, the global joint possibility distribution over the variable set $V = \{A_1, A_2, ..., A_N\}$ can be expressed as the minimum of the N initial a priori and conditional possibilities via the following min-based chain rule:*

$$\pi_m(A_1, .., A_N) = \min_{i=1..N} \Pi(A_i \mid U_{A_i}). \tag{5.6}$$

The min-based chain rule is derived from the (minimum) independence relations induced by the DAG structure and local min-based conditional degrees.

Indeed, as shown in Chapter 3, several definitions of independence are based on the minimum operator. In what follows we use the non-interactivity independence relation (introduced in Section 3.3.2) in different algorithms related to the min-based possibilistic graphs. Recall that the non-interactivity is defined by:

$$\Pi(x \wedge y \mid z) = min(\Pi(x \mid z), \Pi(y \mid z)), \forall x, y, z,$$

to express that the variable sets X and Y are NI-independent in the context Z.
We also recall that in the DAG "*each node is independent of its non-descendants in the context of its parents*".

Therefore, the min-based chain rule can be explained in the following way:
Let $d = (A_1, ..., A_N)$ be an ordering of the variables in V such that $\forall A_i, U_{A_i} \subseteq \{A_{i+1}, ..., A_N\}$.
Let $R_{A_i} = V - U_{A_i}$. Then, we have by definition:
$\pi_m(A_1, ..., A_N) = min(\Pi(A_1, R_{A_1} \mid U_{A_1}), \Pi(U_{A_1}))$
$= min(min(\Pi(A_1 \mid U_{A_1}), \Pi(R_{A_1} \mid U_{A_1})), \Pi(U_{A_1}))$ (since A_1 is a leaf (i.e. $R_{A_1} = Z_{A_1}$) and A_1 is independent of its non-descendants (i.e. Z_{A_1}) in the context of its parent set U_{A_1})
$= min(\Pi(A_1 \mid U_{A_1}), \Pi(R_{A_1} \mid U_{A_1}), \Pi(U_{A_1}))$
$= min(\Pi(A_1 \mid U_{A_1}), min(\Pi(R_{A_1} \mid U_{A_1}), \Pi(U_{A_1})))$
$= min(\Pi(A_1 \mid U_{A_1}), \Pi(A_2, ..., A_N))$ (by definition)
Iterating the same operation on the rest of variables leads to (5.6). Note that this construction is not similar to the product-based one.

Example 5.4 *Let ΠG_m be a min-based possibilistic network defined by three variables A, B and C (see Figure 5.2) and the a priori and conditional possibility distributions given in Table and 5.5. Using the min-based chain rule (5.6) we obtain the joint possibility distribution given in Table 5.6.*

Figure 5.2: Example of a singly connected DAG

Table 5.5: Initial distributions

a	$\Pi(a)$	b	$\Pi(b)$	a	b	c	$\Pi(c \mid a \wedge b)$	a	b	c	$\Pi(c \mid a \wedge b)$
a_1	*1*	b_1	*0.3*	a_1	b_1	c_1	*0.4*	a_2	b_1	c_1	*0.1*
a_2	*0.2*	b_2	*1*	a_1	b_1	c_2	*1*	a_2	b_1	c_2	*1*
				a_1	b_2	c_1	*0.3*	a_2	b_2	c_1	*1*
				a_1	b_2	c_2	*1*	a_2	b_2	c_2	*0.1*

Table 5.6: Global joint distribution using min-based chain rule (5.6)

a	b	c	$\pi_m(a \wedge b \wedge c)$	a	b	c	$\pi_m(a \wedge b \wedge c)$
a_1	b_1	c_1	*0.3*	a_2	b_1	c_1	*0.1*
a_1	b_1	c_2	*0.3*	a_2	b_1	c_2	*0.2*
a_1	b_2	c_1	*0.3*	a_2	b_2	c_1	*0.2*
a_1	b_2	c_2	*1*	a_2	b_2	c_2	*0.1*

The min-based chain rule does not share the same properties than the product-based one. Indeed, it may happen that the joint distribution associated with the network do not recover the initial data. Nevertheless, we show that unrecovered data correspond to redundant data that can be ignored and that they have no influence on independence relations.

5.4.1 Recovering initial data

A min-based possibilistic network ΠG_m is said to be *coherent* if the application of the chain rule (5.6) allows the recovering of the initial data provided by the expert i.e. $\Pi_m(a \mid u_A) = \Pi(a \mid u_A)$ where $\Pi_m(a \mid u_A)$ is the conditional possibility degree computed from π_m and $\Pi(a \mid u_A)$ is the initial degree.

As pointed out by Fonck [62] such equality does not always hold as shown by the following counter-example:

Counter-example 5.1 *Let us reconsider the min-based possibilistic network given in Example 5.4. From Table 5.6, we can check that*
$\Pi_m(c_1 \mid a_1 \wedge b_1) = 1 \neq \Pi(c_1 \mid a_1 \wedge b_1) = 0.4$ *since* $\Pi_m(a_1 \wedge b_1 \wedge c_1) = \Pi_m(a_1 \wedge b_1) = 0.3$. *This is due to the fact that the initial values provided by the expert are not coherent with the axioms of possibility distributions. Indeed, using (1.17) we always have if* $\Pi(p \mid q) \neq 1$ *then* $\Pi(p \mid q) = \Pi(p \wedge q) < \Pi(q)$. *However, in our example* $\Pi(c_1 \mid a_1 \wedge b_1) = 0.4 > \Pi(a_1 \wedge b_1) = 0.3$.

The following proposition goes one step further and compares the exact value of $\Pi_m(a \mid u_A)$ with respect to the initial local distribution i.e. $\Pi(a \mid u_A)$. It generalizes the result given in [16] to the case of non binary variables.

Proposition 5.4 *Let* π_m *be the global joint possibility distribution of* ΠG_m *computed using (5.6). Let* $\Pi(a \mid u_A)$ *be the conditional distribution given by the expert on the node* A *and* $\Pi_m(a \mid u_A)$ *be the conditional possibility computed from* π_m. *Then,*
either $\Pi_m(a \mid u_A) = \Pi(a \mid u_A)$ *or* $\Pi_m(a \mid u_A) = 1$.
Moreover if $\Pi_m(a \mid u_A) = 1 \neq \Pi(a \mid u_A)$, *then* $\Pi(a \mid u_A) > \Pi_m(u_A)$.

The proof of this proposition needs the following technical Lemma [16] (similar to Lemma 5.1).

Lemma 5.2 *Let* X *be a strict subset of* V *and* x *be a fixed instantiation of* X. *Let* $Z = V - X$. *Then,*

$$max_z\{min\{\Pi(a \mid u_A) : a \in z, u_A \subseteq z \wedge x\}\} = 1. \tag{5.7}$$

Proposition 5.4 means that the computed joint distribution either preserves the initial values or push them up to 1. For instance, in Example 5.4, $\Pi_m(c_1 \mid a_1 \wedge b_1)$ is equal to 1 instead of the original value $\Pi(c_1 \mid a_1 \wedge b_1) = 0.4$.

However, this should not be viewed as a drawback. Indeed, this simply means that the experts give useless data which are aggregated by the min-based chain rule with more specific (precise) ones. The following proposition shows that the initial *unrecovered* data can be ignored without any effect on the global joint distribution.

Proposition 5.5 *Let* $A \in V$ *be a variable in* ΠG_m *s.t.* $\Pi_m(a \mid u_A) \neq \Pi(a \mid u_A)$. *Let* π'_m *be a new joint distribution obtained from* ΠG_m *by only substituting the value* $\Pi(a \mid u_A)$ *by the degree 1. Then,*

$$\pi'_m = \pi_m. \tag{5.8}$$

The idea of the proof of this proposition is to show that for any variable A_j with an initial incoherent local distribution (i.e. $\Pi(a_j \mid u_{A_j}) < 1$ and $\Pi(a_j \mid u_{A_j}) > \Pi_m(u_{A_j})$), the joint distribution is equal to the minimum between all initial local distributions except the one relative to A_j (i.e. $\pi_m(v) = \min_{i=1..N, i \neq j} \Pi(a_i \mid u_{A_i})$). Indeed, when considering an ordering $d = (A_1, ..., A_N)$ of the variables in V such that $\forall A_i, U_{A_i} \subseteq \{A_{i+1}, ..., A_N\}$, we prove that the initial incoherent local distribution relative to A_j is greater than the minimum between all initial local distributions relative to $A_{j+1}, .., A_N$ (i.e. $\Pi(a_j \mid u_{A_j}) > \min_{i=j+1..N} \Pi(a_i \mid u_{A_i})$)

Example 5.5 *Let us continue Example 5.1. If we compute the possibility distributions relative to A, B and C by marginalization of the joint distribution given by Table 5.6, we can check that we recover all the initial values except* $\Pi(c_1 \mid a_1 \wedge b_2)$ *since* $\Pi_m(c_1 \mid a_1 \wedge b_2)$ *is equal to 1 instead of 0.4. However, we can recover the original possibility distribution using* $\Pi_m(A), \Pi_m(B)$ *and* $\Pi_m(C \mid A, B)$ *(see Table 5.7).*

Table 5.7: New joint distribution

a	b	c	$\Pi_m(a)$	$\Pi_m(b)$	$\Pi_m(c \mid a \wedge b)$	$\pi'_m(a \wedge b \wedge c)$
a_1	b_1	c_1	*1*	*0.3*	***1***	*0.3*
a_1	b_1	c_2	*1*	*0.3*	*1*	*0.3*
a_1	b_2	c_1	*1*	*1*	*0.3*	*0.3*
a_1	b_2	c_2	*1*	*1*	*1*	*1*
a_2	b_1	c_1	*0.2*	*0.3*	*0.1*	*0.1*
a_2	b_1	c_2	*0.2*	*0.3*	*1*	*0.2*
a_2	b_2	c_1	*0.2*	*1*	*1*	*0.2*
a_2	b_2	c_2	*0.2*	*1*	*0.1*	*0.1*

This behavior also exists in possibilistic logic [16], namely a possibility distribution associated with a possibilistic base do not guarantee to recover the exact value of the knowledge base. for instance, it is enough to consider a knowledge base $\Sigma = \{(a, 0.8), (a \vee b, 0.4)\}$. Then, $\pi_\Sigma(a \wedge b) = \pi_\Sigma(a \wedge \neg b) = 1, \pi_\Sigma(\neg a \wedge b) = 0.2, \pi_\Sigma(\neg a \wedge \neg b) = 0.2$. Thus $N(a \vee b) = 1 - 0.2 = 0.8$. This is due to the fact that $(a \vee b, 0.4)$ is strictly subsumed by $(a, 0.8)$.

5.4.2 Recovering independence relations

We now show that the NI-independence relation (used as independence relation relative to ΠG_m) recovers the whole independence relations implied by the DAG structure, exactly as in Bayesian networks and that the initial *unrecovered* data have no effect on this independence property.

As mentioned in Chapter 4, to satisfy this condition the independence relation should be a semi-graphoid. Moreover, in the joint distribution computed via the chain rule, each variable $A \in V$ should be conditionally independent of *its non-descendants (Z_A) given its parents (U_A)*. These two conditions are satisfied by the NI-independence. Indeed, as mentioned in Chapter 3, this relation is a semi-graphoid [62].

Moreover, In [16] it has been shown that in a ΠG_m composed of binary variables, each variable is NI-independent of each of its non-descendants (Z_A) given its parents, in the joint distribution induced from initial conditional distributions. The following Proposition generalizes this result to the case of non binary variables and to the whole set of non-descendants instead of only one variable pertaining to it.

Proposition 5.6 *Let ΠG_m be a min-based possibilistic network. Let π_m be the joint possibility distribution computed using 5.6. Then, each variable $A \in V$, is NI-independent of the variables in Z_A given its parent set U_A i.e. $\forall a \in D_A, \forall u_A \in D_{U_A}, \forall z_A \in D_{Z_A}$:*

$$\Pi_m(a \wedge u_A \mid z_A) = min(\Pi_m(a \mid u_A), \Pi_m(u_A \mid z_A)). \tag{5.9}$$

Example 5.6 *Let ΠG_m be a min-based possibilistic network defined by the DAG of Figure 5.1 and the initial possibility distributions given in Example 5.1. We can for instance check that D is NI-independent of $Z_D = \{A, B, C\}$ in the context of $U_D = \{A\}$. Indeed,*
$\forall a, b, c, d, \Pi_m(a \wedge b \wedge c \wedge d) = min(\Pi_m(a \wedge d), \Pi_m(a \wedge b \wedge c))$ as shown in Tables 5.8 and 5.9.

Table 5.8: Joint distribution using min-based chain rule

a	b	c	d	$\pi_m(a \wedge b \wedge c \wedge d)$	a	b	c	d	$\pi_m(a \wedge b \wedge c \wedge d)$
a_1	b_1	c_1	d_1	*1*	a_2	b_1	c_1	d_1	*0.2*
a_1	b_1	c_1	d_2	*0*	a_2	b_1	c_1	d_2	*0.2*
a_1	b_1	c_2	d_1	*0.6*	a_2	b_1	c_2	d_1	*0.2*
a_1	b_1	c_2	d_2	*0*	a_2	b_1	c_2	d_2	*0.7*
a_1	b_2	c_1	d_1	*0.3*	a_2	b_2	c_1	d_1	*0.1*
a_1	b_2	c_1	d_2	*0*	a_2	b_2	c_1	d_2	*0.1*
a_1	b_2	c_2	d_1	*0.2*	a_2	b_2	c_2	d_1	*0.2*
a_1	b_2	c_2	d_2	*0*	a_2	b_2	c_2	d_2	*0.3*

Thus, using Proposition 5.6 and Corollary 4.1, we can give the following fundamental proposition:

Proposition 5.7 *Let ΠG_m be a min-based possibilistic network. Let π_m be the joint distribution relative to ΠG_m. Let X, Y and Z be three disjoint subsets of V. If X and Y are*

Table 5.9: Local joint distributions

a	b	c	$\pi_m(a \wedge b \wedge c)$	a	b	c	$\pi_m(a \wedge b \wedge c)$	a	d	$\pi_m(a \wedge d)$
a_1	b_1	c_1	*1*	a_2	b_1	c_1	*0.2*	a_1	d_1	*1*
a_1	b_1	c_2	*0.6*	a_2	b_1	c_2	*0.7*	a_1	d_2	*0*
a_1	b_2	c_1	*0.3*	a_2	b_2	c_1	*0.1*	a_2	d_1	*0.2*
a_1	b_2	c_2	*0.2*	a_2	b_2	c_2	*0.3*	a_2	d_2	*0.7*

d-separated by Z in G, then X and Y are NI-independent in the context of Z in π_m. More formally:

$$< X|Z|Y >_{\mathcal{G}} \Rightarrow I_{NI}(X, Z, Y). \tag{5.10}$$

In other terms, any independence relation implied by the DAG structure of a min-based possibilistic networks ΠG_m, can be recovered from the joint distribution relative to ΠG_m computed via the min-based chain rule (5.6).

It is clear that there is no need to have a coherent network to satisfy this Proposition since the initial unrecovered data have no effect on the value of the joint distribution due to Propositions 5.4 and 5.5.

However, we have to mention that this Proposition is not available if MS-independence[1] is used instead of NI-independence, then we can check that the independence relations implied by the DAG structure are not always recovered, using the min-based chain rule, as shown by the following counter-example:

Counter-example 5.2 *Let us consider the min-based possibilistic network given in Example 5.6. We can, for instance, check that the independence relation $< D|U_D|Z_D >_{\mathcal{G}}$ is not recovered. Indeed, D is not MS-independent of $Z_D = \{A, B, C\}$ in the context of $U_D = \{A\}$ since from the joint distribution given in Table 5.8 we can check that $\Pi_m(d_1 \mid a_2 \wedge b_2 \wedge c_1) = 1 \neq \Pi_m(d_1 \mid a_2) = 0.2$.*

[1] we recall that I_{MS}, presented in Section 3.3.1, is defined by $\Pi(x \mid y \wedge z) = \Pi(x \mid z)$ and $\Pi(y \mid x \wedge z) = \Pi(y \mid z), \forall x, y, z$ to express that the variable sets X and Y are MS-independent in the context Z.

5.5 Conclusion

In this Chapter, we have presented possibilistic counterparts of Bayesian networks. Namely, product-based possibilistic networks and min-based possibilistic networks. We have shown that when we use the product form of conditioning we get possibilistic networks very close to the probabilistic ones.

Besides, networks based on Spohn's ordinal conditional functions [35, 72] and those based on the theory of evidence (in the case of nested focal elements) encode product-based possibilistic networks since they use the same conditioning (as noted in Chapter 1). In addition, Shenoy has treated the case of possibilistic VBS using only the product operator [112].

This means that most of existing works which encodes possibilistic networks use the product operator and hence share same theoretical and practical results than probabilistic networks.

This is not the case with min-based networks which differ from product-based networks since they do not satisfy the coherence property. Indeed, it may happen that the joint distribution associated with the possibilistic network do not recover the initial data provided by the experts. Nevertheless, we have shown that unrecovered data correspond to redundant data which can be ignored and that they have no influence on independence relations.

Chapter 6, proposes an adaptation of probabilistic propagation algorithms to min-based and product-based networks and shows that unrecovered data have no influence on the propagation process too.

The last part of this thesis exploits specific properties of the minimum operator and proposes a new propagation algorithm for min-based networks.

Chapter 6

Possibilistic Adaptation of Probabilistic Propagation Algorithms

6.1 Introduction

The most common task we wish to solve using possibilistic networks is *possibilistic inference or propagation* which corresponds to determine how the realization of specific values of some variables affects the remaining variables. Adaptations of well known probabilistic propagation algorithms have been proposed in [23, 63, 69]. For instance, Fonck has proposed an adaptation of the original version of Pearl's algorithm [63]. In this Chapter, we adapt the centralized version of this algorithm (proposed by Poet and Shachter [105]) to possibilistic networks. Then we study the possibilistic propagation in the more general case of junction trees.

This Chapter is composed of two main sections. Section 6.2 presents propagation in product-based possibilistic networks. Section 6.3 studies propagation in min-based possibilistic networks. Moreover, it shows that in such networks the initial *unrecovered* data have no effect on the propagation process.
Proofs of this Chapter are given in Appendix D.

6.2 Propagation in product-based possibilistic networks

6.2.1 Product-based propagation in polytrees

We now propose a product-based possibilistic adaptation of probabilistic propagation algorithm in polytrees presented in Section 4.6. This algorithm is based on local communication via two kinds of messages, called λ-messages and μ-messages circulating, respectively, from

children to parents and from parents to children. This massage passing is performed in a similar way than in Pearl's algorithm [87, 103].

In the following, we use exactly the same notations than in Chapter 4 (see Section 4.6).

We now give expressions of different messages ($Bel(A)$, $\lambda(A)$, $\mu(A)$, $\lambda_{Y_j}(A)$ and $\mu_A(U_i)$).

- $\forall a \in D_A$, the current conditional possibility measure of a based on the total evidence e is defined by:

$$Bel(a) = \Pi_p(a \mid e) = \alpha \cdot \lambda(a) \cdot \mu(a), \tag{6.1}$$

  where $\alpha = \frac{1}{max_a Bel(a)}$.

- The λ value $\forall a \in D_A$ is defined by:

$$\lambda(a) = \Pi_p(a \mid e_A^-) = \lambda_A(a) \cdot \prod_{j=1}^{m} \lambda_{Y_j}(a), \tag{6.2}$$

  where $\lambda_A(a)$ denotes local evidence related to the node A such that:

$$\lambda_A(a) = \begin{cases} 0 & \text{if } e_A \neq a \text{ (A is instanciated to } (e_A \neq a) \\ 1 & \text{otherwise (A is instanciated to a } (e_A = a) \\ & \text{or A is not instanciated).} \end{cases}$$

- The μ value $\forall a \in D_A$ is defined by:

$$\mu(a) = \Pi_p(a \mid e_A^+) = \max_u \Pi(a \mid u) \cdot \prod_{i=1}^{n} \mu_A(u_i). \tag{6.3}$$

- The λ message from A to its parent $U_i, (i \in \{1, ..n\})$ when $U_i = u_i$ is defined by:

$$\lambda_A(u_i) = \Pi_p(e_{U_i A}^- \mid u_i) = \beta \max_a \lambda(a) [\max_{u_k : k \neq i} \Pi(a \mid u) \cdot \prod_{k \neq i} \mu_A(u_k)], \tag{6.4}$$

  where β is a normalization constant.

  If the graph is a rooted tree (each node has a unique parent), then this message is simplifies to:

$$\lambda_A(u_i) = \beta \max_a \lambda(a) \cdot \Pi(a \mid u).$$

- The μ message from A to its child $Y_j, (j \in \{1, ..m\})$ when $A = a$ is defined by:

$$\mu_{Y_j}(a) = \Pi_p(a \mid e_{AY_j}^+) = \alpha \cdot \lambda_A(a) \cdot \prod_{i=1..m, i \neq j} \lambda_{Y_i}(a) \cdot \mu(a). \tag{6.5}$$

Note that these formulas are similar to those corresponding to Bayesian networks presented in Section 4.6 but use the maximum operator instead of the addition.

Propagation algorithm

The first step is to select an arbitrary node within the smallest connected set of nodes containing all of the modified nodes (denoted by $\mathcal{S}$), to be the *pivot* node. Then, a message passing starts via two phases:

- A *collect-evidence* phase, in which each node in $\mathcal{S}$ passes a message to its adjacent nodes in the pivot direction (each node has only one adjacent node in the pivot direction), beginning with the node farthest from the pivot in $\mathcal{S}$.
- A *distribute-evidence* phase, in which each node in the graph passes messages to its adjacent nodes away from the pivot direction, beginning with the pivot itself until the leaves in the DAG are reached.

The outline of this algorithm is as follows:

Algorithm 6.1: Product-based propagation in polytrees

begin

Choosing the root of propagation
- Let $\mathcal{S}$ be the smallest connected set which contains the observed nodes;
- Let *Pivot* be an arbitrary node within $\mathcal{S}$ representing the root of propagation;
- Let *Postorder* be the vector containing the order in which messages are sent in the distribute-evidence phase (the last node is the pivot);
- Let *Preorder* be the vector containing the order in which messages are sent in the collect-evidence phase (the first node is the pivot);

Initialization
- Set all λ and μ values and messages to 1;
- For each root A, $\mu(a) \leftarrow \Pi(a), \forall a \in D_A$;
- For each observed node A, set $\lambda_A(a)$ to 1 if A is instanciated to a and to 0 otherwise;

Collect-evidence
for *i ← 1 to length(Postorder)-1* **do**
  $A \leftarrow$ *Postorder*[i];
  $B \leftarrow$ adjacent node of A in *Postorder*);
  Compute $\lambda(A)$ using (6.2), Compute $\mu(A)$ using (6.3);
  if B *is a parent of* A **then** post a λ message from A to B using (6.4) **else** post a μ message from A to B using (6.5)

Distribute-evidence
for *i ← 1 to length(Preorder)* **do**
  $A \leftarrow$ *Preorder*[i];
  Below ← adjacent node of A in *Preorder*;
  Compute $\lambda(A)$ using (6.2), Compute $\mu(A)$ using (6.3)
  for *j ← 1 to length(*Below*)* **do**
    B ← *Below*[j];
    if B *is a parent of* A **then** post a λ message from A to B using (6.4) **else** post a μ message from A to B using (6.5)

Marginalization
For each node A, compute $Bel(A) = \Pi_p(a \mid e)$ using (6.1);

end

This algorithm is developed with the same complexity as the probabilistic one i.e. $O(s*N)$ where s is the size of the largest conditional possibility table and N the number of variables. Indeed, this adaptation is also based on a two message passing and ensures that in each direction only one message need to be sent on any arc, thus it converges in two iterations (one for the collect and the other for the distribution) and the number of messages is less than twice the number of nodes.

6.2.2 Product-based propagation in multiply connected DAGs

The principle of this propagation method is similar to the probabilistic propagation in junction trees presented in Chapter 4. Indeed, it is based on the transformation of the initial DAG into a *junction tree* which will be used during the propagation process.

The proposed adaptation has the same complexity than the probabilistic case and remains *NP-complete* since the transformation step of the initial DAG into a junction tree remains the same.

Junction trees (denoted by $\mathcal{JT}$) are defined and constructed in the same manner than in the probabilistic case (see Section 4.7) since this procedure is completely independent of the numerical values.

For each cluster C_i (resp. separator S_{ij}) of $\mathcal{JT}$ we assign a local joint distribution, called *potential* and denoted by π_{C_i} (resp. $\pi_{S_{ij}}$).

From $\mathcal{JT}$, we can associate a unique global joint possibility distribution denoted by, $\pi_{\mathcal{JT}}$ defined by:

Definition 6.1 *The joint distribution associated with $\mathcal{JT}$ is expressed by:*

$$\pi_{\mathcal{JT}}(A_1, .., A_N) = \frac{\prod_{i=1}^{m} \pi_{C_i}}{\prod_{j=1}^{m-1} \pi_{S_{ij}}}, \tag{6.6}$$

where m is the number of clusters in $\mathcal{JT}$.

We now give some definitions regarding to junction trees:

Definition 6.2 *Let C_i and C_j be two adjacent clusters in a junction tree $\mathcal{JT}$ and let S_{ij} be their separator. Then, the link between C_i and C_j is said to be **stable** or **consistent** if:*

$$\max_{C_i \setminus S_{ij}} \pi_{C_i} = \pi_{S_{ij}} = \max_{C_j \setminus S_{ij}} \pi_{C_j}, \tag{6.7}$$

where $\max_{C_i \setminus S_{ij}} \pi_{C_i}$ *is the marginal distribution of* S_{ij} *defined from* π_{C_i}.

If all links in a junction tree are consistent, then the junction tree is said to be ***globally consistent***.

The following proposition shows that when a junction tree is globally consistent, then the potential of each cluster corresponds to its local joint distribution computed from the initial network.

Proposition 6.1 *If a junction tree is globally consistent, then the potential of each cluster* C_i *satisfies:*

$$\pi_{C_i} = \Pi_p(C_i). \tag{6.8}$$

Using this proposition, we can compute the possibility distribution of any variable $A \in V$ in a globally consistent junction tree, using any cluster C_i containing A by marginalizing its potential on A as follows:

$$\Pi_p(A) = \max_{C_i \setminus A} \pi_{C_i}. \tag{6.9}$$

In the following, we denote by $\pi^t_{C_i}$ the potential of the cluster C_i at a step t of the propagation. $t = I$ (resp. $t = C$) corresponds to the initialization (resp. global consistency) step.

Possibilistic propagation

Once the transformation of the DAG into a junction tree is performed, the propagation process starts and it will be possible to compute for any variable $A \in V$, the possibility degree $\Pi_p(A)$. The more general problem of computing $\Pi_p(A \mid e)$, where e is the total evidence, is addressed is addressed later. We now present the principle steps of possibility propagation in junction trees with no evidence.

Step 1: Initialization. The first step in the propagation is to initialize the junction tree. In this phase we quantify the junction tree using initial possibility distributions as follows:

Algorithm 6.2: Initialization

begin

For each cluster $C_i : \pi^I_{C_i} \leftarrow 1$;
For each separator $S_{ij} : \pi^I_{S_{ij}} \leftarrow 1$;
For each variable A choose a cluster C_i containing $\{A\} \cup U_A$:
$\pi^I_{C_i} \leftarrow \pi^I_{C_i} \cdot \Pi(A \mid U_A)$;

end

The following proposition, shows that the initialized junction tree encodes the same distribution than the initial network.

Proposition 6.2 *Let ΠG_p be a product-based possibilistic network. Let $\mathcal{JT}$ be the junction tree corresponding to ΠG_p generated by the above initialization procedure. Let π_p be the joint distribution encoded by ΠG_p (using (5.1)) and $\pi^I_{\mathcal{JT}}$ be the joint distribution encoded by $\mathcal{JT}$ (using (6.6)). Then,*

$$\pi_p = \pi^I_{\mathcal{JT}}. \tag{6.10}$$

The initialized junction tree can be *inconsistent* since this initial assignment does not guarantee the global consistency requirement of Equation (6.7). Thus, we should run the second step of global propagation which ensures global consistency.

Step 2: Global propagation. Once the junction tree is initialized, the global propagation is performed in order to make it globally consistent. The global propagation is performed via a *message passing* mechanism between each cluster C_i and its adjacent cluster C_j divided into two phases starting from an arbitrary cluster as a pivot node:

- A *collect-evidence* phase in which each cluster passes a message to its adjacent cluster in the pivot direction, beginning with the clusters farthest from the pivot. In this phase messages are starting from the leaves. The order in which messages are sent is denoted by the *postorder* of the pivot.
- A *distribute-evidence* phase in which each cluster passes messages to its adjacent clusters away from the pivot direction, beginning with the pivot itself. In this phase messages circulate from the pivot until the leaves are reached. The order in which messages are sent is denoted by the *preorder* of the pivot.

In both phases if a cluster C_i sends a message to its adjacent cluster C_j, then the potentials of C_i, C_j and their separator S_{ij} are updated as follows:

1. Save the same potential for C_i
$$\pi^{t+1}_{C_i} \leftarrow \pi^t_{C_i}. \tag{6.11}$$
2. Assign a new potential to S_{ij}
$$\pi^{t+1}_{S_{ij}} \leftarrow \max_{C_i \setminus S_{ij}} \pi^t_{C_i}. \tag{6.12}$$
3. Assign a new potential to C_j:
$$\pi^{t+1}_{C_j} \leftarrow \pi^t_{C_j} \cdot \frac{\pi^{t+1}_{S_{ij}}}{\pi^t_{S_{ij}}}. \tag{6.13}$$

The outline of the global propagation procedure is as follows:

Algorithm 6.3: Global propagation

begin
Choosing the root of propagation
- Let *Pivot* be an arbitrary cluster C_i to represent the root of propagation;
- Let *Postorder* be the vector containing the order in which messages are sent in the distribute-evidence phase (the last node is the pivot);
- Let *Preorder* be the vector containing the order in which messages are sent in the collect-evidence phase (the first node is the pivot);
Collect-evidence
for *i ← 1 to length(Postorder)-1* **do**
  $C_i \leftarrow$ *Postorder*[i];
  $C_j \leftarrow$ adjacent cluster of C_i in *Postorder*;
  Post a message from C_i to C_j using (4.13), (4.14) and (4.15);
Distribute-evidence
for *i ← 1 to length(*Preorder*)* **do**
  $C_i \leftarrow$ *Preorder*[i];
  Below ← adjacent clusters of C_i in *Preorder*;
  for *j ← 1 to length(Below)* **do**
    $C_j \leftarrow$ Below[j];
    Post a message from C_i to C_j using (4.13), (4.14) and (4.15);
end

The following proposition shows that at each level of the global propagation procedure, the junction tree encodes the same joint distribution:

Proposition 6.3 *Let $\pi^t_{\mathcal{JT}}$ be the joint distribution relative to a junction tree $\mathcal{JT}$ at level t. Let $\pi^{t+1}_{\mathcal{JT}}$ be the resulted joint distribution after the modification of a cluster C_i using the above procedure. Then,*

$$\pi^t_{\mathcal{JT}} = \pi^{t+1}_{\mathcal{JT}}. \tag{6.14}$$

From Propositions 6.2 and 6.3 we deduce that from the initialization to the global consistency level, the junction tree encodes the same joint distribution.

Proposition 6.4 *Let π_p be the joint distribution encoded by ΠG_p (using (5.1)). Let $\pi^C_{\mathcal{JT}}$ be the joint distribution encoded by $\mathcal{JT}$ after the global propagation procedure (using (6.6)). Then,*

$$\pi_p = \pi^C_{\mathcal{JT}}. \tag{6.15}$$

The following proposition shows that the collect and distribute phases are enough to make the junction tree globally consistent.

Proposition 6.5 *The global consistency is reached after the collect and distribute phases.*

Step 3: Marginalization. Using the consistent junction tree obtained from the previous phase, we can now compute for each variable of interest A, the possibility measure $\Pi_p(A)$ as follows:

Algorithm 6.4: Marginalization

begin
Identify a cluster C_i containing A;
Compute $\Pi_p(A)$ by marginalization of $\pi^C_{C_i}$ on A: $\Pi_p(A) \leftarrow \max_{C_i \setminus A} \pi^C_{C_i}$;
end

Handling the evidence

We now show how to generalize the above propagation algorithm in order to compute for any variable $A \in V$, $\Pi_p(A \wedge e)$ where e is the total evidence. This evidence will be encoded by using a *likelihood* Λ defined by:

$$\Lambda_A(a) = \begin{cases} 1 & \text{if A is not instanciated} \\ 1 & \text{if A is instanciated for a} \\ 0 & \text{if A is instanciated but not for a.} \end{cases} \tag{6.16}$$

To handle the evidence e, we should extend the propagation procedure by transforming the initialization step so that to incorporate any certain information. Indeed, we should encode the evidence e as a likelihood (using (6.16)), then, we incorporate it into the junction tree by adding these two steps to the initialization procedure:
- For any instanciated variable A, encode the observation $A = a$ as a likelihood Λ_A using (6.16).
- Identify a cluster C_i containing A: $\pi^I_{C_i} \leftarrow \pi^I_{C_i} \cdot \Lambda_A$.

By entering the observation set, the junction tree encodes $\Pi(V \wedge e)$ instead of $\Pi(V)$, and all subsequent probabilities derived from it are probabilities of events that are conjoined with evidence e.

Through global propagation and under the assumption that we have an evidence e, the potential of cluster C_i encodes $\Pi_p(C_i \wedge e)$. Then, when we marginalize any cluster potential $\pi^C_{C_i}$ into a variable A (s.t $A \subseteq C_i$) using (6.9), we obtain the possibility measure of A and e:

$$\Pi_p(A \wedge e) = \max_{C_i \setminus A} \pi^C_{C_i}. \tag{6.17}$$

However, our goal is to compute $\Pi_p(A \mid e)$, this value can be easily obtained from $\Pi_p(A \wedge e)$ by applying the definition of product-based conditioning as follows:

$$\Pi_p(A \mid e) = \frac{\Pi_p(A \wedge e)}{\Pi_p(e)} = \frac{\Pi_p(A \wedge e)}{\max_A \Pi_p(A \wedge e)}. \tag{6.18}$$

6.3 Propagation in min-based possibilistic networks

In this Section, we first present an adaptation of the centralized version of Pearl's propagation algorithm. We show that the initial unrecovered data have no effect on the propagation process. Then, we present an adaptation of the probabilistic junction tree propagation algorithm.

6.3.1 Min-based propagation in polytrees

We now propose a min-based possibilistic possibilistic adaptation of probabilistic propagation algorithm in polytrees presented in Section 4.6. This adaptation is slightly different from the one proposed for product-based networks. More precisely, it needs one additional step which concerns the transformation of initial conditional distributions into local joint ones.

From graphs with conditionals to graphs with local distributions

The use of the conditional distributions is not appropriate when we want to derive the expression of the updating messages. Then, an alternative way is to use joint distributions of each variable in the context of its parents. This is possible since the NI-independence can be expressed in a conditional or joint form (see (3.14) and (3.15)). Thus, the question is *how to proceed in order to transform the original conditional possibilities into joint ones* ?

The global joint distribution computed via the min-based chain rule (5.6) depends only on the values which are less than the degree 1 (since we use the min operator). Moreover, (1.17) implies that for each variable $A_i, \forall a_i \in D_{A_i}, \forall u_{A_i} \in D_{U_{A_i}}$:

$$\Pi(a_i \mid u_{A_i}) = \begin{cases} \Pi(a_i \wedge u_{A_i}) & \text{if } \Pi(a_i \mid u_{A_i}) < \Pi(u_{A_i}) \\ 1 & \textit{otherwise.} \end{cases}$$

Then, in the transformation process we can only consider the conditionals which are less than 1 and ignore the rest i.e. the conditionals equal to 1 by maintaining their initial values. Namely, we preserve the initial conditional values by considering them as joint local distributions. Once the transformation is performed, one can compute the global joint possibility distribution over the variable set V based on the local joint distributions derived from the initial conditional ones using the following chain rule which is equivalent to (5.6):

$$\pi_m(A_1, .., A_N) = \min_{i=1}^{N} \Pi(A_i \wedge U_{A_i}). \quad (6.19)$$

Corollary 6.1 *The joint distribution obtained from initial a priori and conditional distributions using (5.6) is equivalent to the one obtained from derived local joint distributions using (6.19).*

The proof of this corollary is immediate since for each variable A_i, we replace $\Pi(a_i \mid u_{A_i})$ by $\Pi(a_i \wedge u_{A_i})$. However, it is not possible to encode the uncertainty, in a direct manner, by local joint distributions of each node in the context of its parents. Indeed, we can lose some independence relations as shown by the following counter-example.

Counter-example 6.1 *Let us consider the min-based possibilistic network ΠG_m, composed of the DAG of Figure 5.2 and the initial possibility distributions given in Table 6.1. Using (6.19), we can compute the global joint distribution given in Table 6.2.*

Table 6.1: Initial distributions

a	$\Pi(a)$	b	$\Pi(b)$	a	b	c	$\Pi(a \wedge b \wedge c)$	a	b	c	$\Pi(a \wedge b \wedge c)$
a_1	*1*	b_1	*1*	a_1	b_1	c_1	*0.5*	a_2	b_1	c_1	*1*
a_2	*1*	b_2	*1*	a_1	b_1	c_2	*0.5*	a_2	b_1	c_2	*1*
				a_1	b_2	c_1	*1*	a_2	b_2	c_1	*1*
				a_1	b_2	c_2	*1*	a_2	b_2	c_2	*1*

Table 6.2: Global joint distribution

a	b	c	$\pi_m(a \wedge b \wedge c)$	a	b	c	$\pi_m(a \wedge b \wedge c)$
a_1	b_1	c_1	*0.5*	a_2	b_1	c_1	*1*
a_1	b_1	c_2	*0.5*	a_2	b_1	c_2	*1*
a_1	b_2	c_1	*1*	a_2	b_2	c_1	*1*
a_1	b_2	c_2	*1*	a_2	b_2	c_2	*1*

From the DAG structure it is clear that A and B are independent. However from the global joint distribution given in Table 6.2 these two variables are not NI-independent since $\Pi_m(a_1 \wedge b_1) = 0.5 \neq min(\Pi_m(a_1), \Pi_m(b_1)) = 1$.

We now give some notations and technical lemmas relative to independence relations in min-based possibilistic networks and useful in the development of the propagation algorithm.

Independence relations

In the following, we use exactly the same notations than in Chapter 4 (see Section 4.6).

Proposition 6.6 summarizes independence relations in a singly connected min-based network (based on NI-independence relation) regarding a particular node A. This proposition is needed to develop different expressions of values and messages used in the propagation process.

Proposition 6.6 ***(Independence relations in a singly connected min-based network)***

1: *In the context of any node A, the two sets E_A^+ and E_A^- are NI-independent.*

2: *$\forall Y_i \in Y$, the node A d-separates $E_{AY_i}^-$ from $\{E_{AY_{i+1}}^-, ..., E_{AY_m}^-\}$ (see Figure 6.1).*

3: *The set E_A^+ and the node A are NI-independent in the context U.*

4: *$\forall U_i \in U, U_i$ and $\{U_{i+1}, ..., U_n\}$ are NI-independent in the context E_A^+ (see Figure 6.2).*

5: *$\forall U_i \in U, U_i$ and $\{E_{U_{i+1}A}^+, ..., E_{U_nA}^+\}$ are NI-independent in the context $E_{U_iA}^+$ (see Figure 6.3).*

6: *$\forall U_i \in U$, $E_{U_iA}^+$ is d-separated from $\{E_{U_{i+1}A}^+, ..., E_{U_nA}^+\}$.*

7: *Given a node A with two parents U_i and V (see Figure 6.4), then the two sets E_{VA}^+ and E_A^- are NI-independent in the context of the nodes $\{A, U_i, V\}$.*

8: *Given a node A with two parents U_i and V (see Figure 6.5), then the two sets E_A^- and $U = \{U_i, V\}$ are NI-independent in the context of A.*

9: *Given a node A with two parents U_i and V, then the two sets E_{VA}^+ and $\{U_i, A\}$ are NI-independent in the context of V.*

Figure 6.1: A d-separates $E_{AY_i}^-$ from $\{E_{AY_{i+1}}^-, ..., E_{AY_m}^-\}$

Figure 6.2: U_i and $\{U_{i+1}, ..., U_n\}$ are NI-independent in the context E_A^+

Figure 6.3: U_i and $\{E^+_{U_{i+1}A}, ..., E^+_{U_nA}\}$ are NI-independent in the context $E^+_{U_iA}$

Figure 6.4: E^+_{VA} and E^-_A are NI-independent in the context of $\{A, U_i, V\}$

Computing the messages

We now give the expression of different values and messages used during the propagation process described later.

Our goal is to compute for each node A its conditional possibility measure based on the total evidence. In other terms, for each instance a of A, we should compute $Bel_{Cdt}(a) = \Pi_m(a \mid e)$. This value can be easily obtained from the joint distribution $Bel_{Joint}(a) = \Pi_m(a \wedge e)$. Thus, we will first compute $Bel_{Joint}(a)$.

Lemma 6.1 ***(Computing joint possibility measure)***
$\forall a \in D_A$, the current joint possibility measure of a based on the total evidence e is defined by:

$$Bel_{Joint}(a) = \Pi_m(a \wedge e) = min(\lambda(a), \mu(a)) \tag{6.20}$$

where $\lambda(a) = \Pi_m(a \wedge e^-_A)$ and $\mu(a) = \Pi_m(a \wedge e^+_A)$.

Then $\forall a \in D_A$, we can compute $Bel_{Cdt}(a) = \Pi_m(a \mid e)$ using $Bel_{Joint}(a) = \Pi_m(a \wedge e)$ by applying the definition of min-based conditioning (1.17) as follows:

$$Bel_{Cdt}(a) = \Pi_m(a \mid e) = \begin{cases} \Pi_m(a \wedge e) & \text{if } \Pi_m(a \wedge e) < \Pi_m(e) = \max_{a \in D_A} \Pi_m(a \wedge e) \\ 1 & \text{otherwise.} \end{cases} \tag{6.21}$$

The expression of $Bel_{Joint}(a)$, depends on the λ-*value* and μ-*value* relative to the node A (denoted, respectively, by $\lambda(A)$ and $\mu(a)$) which depend on the λ-*messages* received from its children and μ-*messages* received from its parents (denoted, respectively, by $\lambda_{Y_j}(A)$ and

Figure 6.5: E^-_A and $U = \{U_i, V\}$ are NI-independent in the context of A

$\mu_A(U_i)$ where $\lambda_{Y_j}(A)$ is the message that A receives from its child Y_j and $\mu_A(U_i)$ is the message that A receives from its parent U_i). We now detail the expression of $\lambda(A)$ and $\mu(a)$.

Lemma 6.2 ***(Computing λ-value)***
The λ value $\forall a \in D_A$ is defined by:

$$\lambda(a) = \Pi_m(a \wedge e_A^-) = min(\lambda_A(a), \min_{j=1}^{m} \lambda_{Y_j}(a)). \tag{6.22}$$

where $\min_{j=1}^{m} \lambda_{Y_j}(a))$ corresponds to the minimum between the λ-messages received from the children of A and $\lambda_A(a)$ denotes local evidence related to the node A such that:

$$\lambda_A(a) = \begin{cases} 0 & \text{if } e_A \neq a \text{ (A is instanciated to } (e_A \neq a) \\ 1 & \text{otherwise (A is instanciated to } a \ (e_A = a) \\ & \text{or } A \text{ is not instanciated).} \end{cases}$$

Lemma 6.3 ***(Computing μ-value)***
The μ value $\forall a \in D_A$ is defined by:

$$\mu(a) = \Pi_m(a \wedge e_A^+) = \max_u min(\Pi(a \wedge u), \min_{i=1}^{n} \mu_A(u_i)). \tag{6.23}$$

where $\min_{i=1}^{n} \mu_A(u_i))$ corresponds to the minimum between the μ-messages received from the parents of A.

The following lemmas give the expressions of λ-messages received from children and μ-messages received from parents.

Lemma 6.4 ***(Computing λ-message)***
The λ message from A to its parent $U_i, (i \in \{1, ..n\})$ when $U_i = u_i$ is defined by:

$$\lambda_A(u_i) = \Pi_m(e_{U_iA}^- \wedge u_i) = \max_{a \in D_A} min[\lambda(a), \max_{u_k : k \neq i} (min(\Pi(a \wedge u), \min_{k \neq i} \mu_A(u_k)))]. \tag{6.24}$$

If the graph is a rooted tree (each node has a unique parent), then this message is simplifies to:

$$\lambda_A(u_i) = \max_a min[\lambda(a), \Pi(a \wedge u)].$$

Lemma 6.5 ***(Computing μ-message)***
The μ message from A to its child $Y_j, (j \in \{1, ..m\})$ when $A = a$ is defined by:

$$\mu_{Y_j}(a) = \Pi_m(a \wedge e_{AY_j}^+) = min(\lambda_A(a), \min_{i=1..m, i \neq j} \lambda_{Y_i}(a), \mu(a)). \tag{6.25}$$

Note that these expressions are similar to those corresponding to Bayesian networks [103] but use the maximum (resp. minimum) operator instead of the addition (resp. product). Moreover, they are based on initial joint distributions rather than conditional ones. Contrary to the probabilistic case all theses messages are *sub-normalized* except Bel_{Cdt}.

Propagation algorithm

The propagation algorithm is basically similar to Algorithm 6.1 by using (6.20) (reps. (6.22), (6.23), (6.24), (6.25)) instead of (6.1) (reps. (6.2), (6.3), (6.4), (6.5)). However, we should use an additional step in the initialization phase in order to transform initial conditional distributions into joint ones. We should also add a supplementary step of *normalization* (after the marginalization) which allows to compute $Bel_{Cdt}(A)$ from $Bel_{Joint}(A)$ using (6.21).

This algorithm is developed with the same complexity as the probabilistic one i.e. $O(s*N)$ where s is the size of the largest conditional possibility table and N the number of variables. Indeed, this adaptation it is also based on a two message passing and ensures that in each direction only one message need to be sent on any arc, thus it converges in two iterations (one for the collect and the other for the distribution) and the number of messages is less than twice the number of nodes.

Effect of initial unrecovered data

In the previous Chapter, we have shown that the initial unrecovered data have no influence neither on the global joint distribution obtained from the min-based chain rule, nor on the independence relations. In the following we show that these values have no effect either on the propagation process.

Indeed, the propagation is equivalent to applying the min-based conditioning on the global joint distribution, taking into account the total evidence, in order to compute the posterior distribution relative to each variable. This is equivalent to say that the interesting values in the propagation process is the joint distribution, which is independent on the initial unrecovered data as stated by proposition 5.5, and the total evidence.

Example 6.1 *Let us reconsider the min-based possibilistic graph ΠG_m given in Example 7.1 and suppose that we receive a certain information about B (i.e. $B = b_1$), then the initial joint distribution π_m (see Table 6.3) obtained from the min-based chain rule (5.6) is transformed into π'_m (see Table 6.3). Then, from π'_m, we can compute the impact of the certain information on A and C: $Bel_{Joint}(A)$=[0.3 0.2], $Bel_{Joint}(C)$=[0.3 0.3]. Then, using these values we can compute the conditional possibilities using (1.17): $Bel_{Cdt}(A)$=[1 0.2], $Bel_{Cdt}(C)$=[1 1].*

The same phenomena is observed with our propagation algorithm. Indeed, each variable A will compute at least one time its μ value defined by (6.23). It is clear that the value of this message is based on the *minimum* between the initial conditional distribution relative to A

Table 6.3: Joint distribution with evidence ($B = b_1$)

a	b	c	$\pi_m(a \wedge b \wedge c)$	$\pi'_m(a \wedge b \wedge c \wedge e)$
a_1	b_1	c_1	*0.3*	*0.3*
a_1	b_1	c_2	*0.3*	*0.3*
a_1	b_2	c_1	*0.3*	*0*
a_1	b_2	c_2	*1*	*0*
a_2	b_1	c_1	*0.1*	*0.1*
a_2	b_1	c_2	*0.2*	*0.2*
a_2	b_2	c_1	*0.2*	*0*
a_2	b_2	c_2	*0.1*	*0*

(since we take $\Pi(a \wedge u) = \Pi(a \mid u)$) and the messages obtained from its parents ($min_{i=1}^{n} \mu_A(u_i)$).

To illustrate this relation let us consider that A has a single parent B which is a non instanciated root then $\mu(a) = \min(\Pi(a \wedge b), \mu_A(b))$.
Moreover, from (6.25) we have, $\mu_A(b) = \min(\lambda_B(b), \mu(b)) = \min(1, \Pi(b)) = \Pi(b)$.
Indeed, $\lambda_B(b) = 1$ since B is non instanciated and $\mu(b) = \Pi(b)$ since it is a root.

This implies that $\mu(a) = \min(\Pi(a \wedge b), \Pi(b))$. Then, if the initial conditional distribution on A does not respect the axioms of possibility distribution i.e.
$\exists a, \exists b$ s.t $\Pi(a \mid b) \neq 1 > \Pi(b)$, then $\Pi(a \wedge b) > \Pi(b)$ (since we take $\Pi(a \wedge u) = \Pi(a \mid u)$).
which implies that $\mu(a) = \min(\Pi(a \wedge b), \Pi(b)) = \Pi(b)$
This means that the unrecovered initial data (here $\Pi(a \wedge b)$) are eliminated by the minimum operator in (6.23) and has no effect in the propagation process.

Example 6.2 *Let us apply our propagation algorithm in the min-based possibilistic network ΠG_m given in Example 6.1:*

- *Choosing the root : $\mathcal{S} \leftarrow \{B\}$, Pivot $\leftarrow B$, Postorder $\leftarrow [B]$, Preorder $\leftarrow [B, C, A]$*
- *Initialization*
  1. *Transformation of $\Pi(C \mid A, B)$ into $\Pi(C, A, B)$*
  2. *Set all λ and μ values and messages to 1*
  3. *The root nodes are A and B, then: $\mu(A)$=[1 0.2], $\mu(B)$=[0.3 1]*
  4. *The observed node is B ($B = b_1$) then: $\lambda_B(B)$=[1 0]*
- *Collect-evidence: We can escape this phase since the only instanciated variable is the pivot*

- *Distribute-evidence: The messages are passed from the pivot node to the rest of the network with the following values:*

  1. Preorder[1] =B: $\mu(B)$*=[0.3 1],* $\lambda(B)$*=[1 0]*

  Post a μ *message from B to C:* $\lambda_C(B)$*=[0.3 0]*

  2. Postorder[2] =C: $\mu(C)$*=[0.3 0.3],* $\lambda(C)$*=[1 1]*

  Post a λ *message from C to A:* $\lambda_C(A)$*=[0.3 0.3]*

  3. Postorder[3] =A: $\mu(A)$*=[1 0.2],* $\lambda(A)$*=[0.3 0.3]*

- *Marginalization:* $Bel_{Joint}(A)$*=[0.3 0.2],* $Bel_{Joint}(C)$*=[0.3 0.3]*

- *Normalization:* $Bel_{Cdt}(A)$*=[1 0.2],* $Bel_{Cdt}(C)$*=[1 1].*

Note that we find the same values than in Example 6.1. It is clear that initial unrecovered distribution $\Pi(c_1 \mid a_1 \wedge b_1)$ *(see Example 5.1) has no effect on updated values since it has been eliminated when computing* $\mu(C)$*. Indeed:*
$\mu(c_1) = max[min(\Pi(a_1, b_1, c_1), min(\mu_C(a_1), \mu_C(b_1))), min(\Pi(a_1, b_2, c_1), min(\mu_C(a_1), \mu_C(b_2))),$
$min(\Pi(a_2, b_1, c_1), min(\mu_C(a_2), \mu_C(b_1))), min(\Pi(a_2, b_2, c_1), min(\mu_C(a_2), \mu_C(b_2)))]$
$= max[min(0.4, min(1, 0.3)), min(0.1, min(1, 0)), min(0.3, min(1, 0.3)), min(1, min(1, 0))]$
$= max[0.3, 0, 0.3, 0] = 0.3$

6.3.2 Min-based propagation in multiply connect DAGs

The principle of this propagation method is similar to the probabilistic propagation in junction trees presented in Chapter 4. Indeed, it is based on the transformation of the initial DAG into a *junction tree* which will be used during the propagation process.

Junction trees relative to min-based possibilistic networks are defined and constructed in the same manner than in product-based networks (see Section 6.2.2).

The proposed adaptation has the same complexity than the probabilistic case and remains *NP-complete* since the transformation step of the initial DAG into a junction tree remains the same. In Chapter 7, we propose a new algorithm which avoids this step.

Given a junction tree $\mathcal{JT}$, we can compute a unique global joint possibility distribution denoted by, $\pi_{\mathcal{JT}}$ defined by:

Definition 6.3 *The joint distribution associated with* $\mathcal{JT}$ *is expressed by:*

$$\pi_{\mathcal{JT}}(A_1, .., A_N) = \min_{i=1..N} \pi_{C_i}, \tag{6.26}$$

where m is the number of clusters in $\mathcal{JT}$.

Proposition 6.7 shows that when a junction tree is globally consistent[1], then the potential of each cluster corresponds to its local joint distribution computed from the initial network.

Proposition 6.7 *If a junction tree is globally consistent, then the potential of each cluster C_i satisfies:*

$$\pi_{C_i} = \Pi_m(C_i). \tag{6.27}$$

Using this proposition, we can compute the possibility distribution of any variable A in a globally consistent junction tree, using any cluster C_i containing A by marginalizing its potential on A, as follows:

$$\Pi_m(A) = \max_{C_i \setminus A} \pi_{C_i}. \tag{6.28}$$

Possibilistic propagation

Once the transformation of the DAG into a junction tree is performed, the propagation process can start and it will be possible to compute for any variable $A \in V$, the possibility degree $\Pi_m(A)$. The more general problem of computing $\Pi_m(A \mid e)$, where e is the total evidence, is addressed in Section later. We now present the principle steps of possibility propagation in junction trees with no evidence.

Step 1: Initialization. The first step in the propagation is to initialize the junction tree. In this phase we quantify the junction tree using initial possibility distributions as follows:

Algorithm 6.5: Initialization

begin
For each cluster $C_i : \pi^I_{C_i} \leftarrow 1$;
For each separator $S_{ij} : \pi^I_{S_{ij}} \leftarrow 1$;
For each variable A choose a cluster C_i containing $\{A\} \cup U_A$:
$\pi^I_{C_i} \leftarrow min(\pi^I_{C_i}, \Pi(A \mid U_A))$;
end

The following proposition, shows that the initialized junction tree encodes the same distribution than the initial network.

Proposition 6.8 *Let ΠG_m be a min-based possibilistic network. Let $\mathcal{JT}$ be the junction tree corresponding to ΠG_m using the above initialization procedure. Let π_m be the joint distribution*

[1] consistency is defined in the same manner than in product-based junction trees (see Definition 6.2).

encoded by ΠG_m *(using (5.6)) and* $\pi^I_{\mathcal{JT}}$ *be the joint distribution encoded by* $\mathcal{JT}$ *(using (6.26)). Then,*

$$\pi_m = \pi^I_{\mathcal{JT}}. \tag{6.29}$$

The output of this step can be an *inconsistent* junction tree since this initial assignment does not guarantee the global consistency requirement. Thus, we should run the second step of global propagation which ensures global consistency.

Step 2: Global propagation. Once the junction tree is initialized, the global propagation is performed in order to make it globally consistent. The global propagation is performed via a *message passing* mechanism between each cluster C_i and its adjacent cluster C_j in the same manner than in probabilistic junction trees.

If a cluster C_i sends a message to its adjacent cluster C_j, then the potentials of C_i, C_j and their separator S_{ij} are updated as follows:

1. Save the same potential for C_i
$$\pi^{t+1}_{C_i} \leftarrow \pi^t_{C_i}. \tag{6.30}$$
2. Assign a new potential to S_{ij}
$$\pi^{t+1}_{S_{ij}} \leftarrow \max_{C_i \setminus S_{ij}} \pi^t_{C_i}. \tag{6.31}$$
3. Assign a new potential to C_j:
$$\pi^{t+1}_{C_j} \leftarrow min(\pi^t_{C_j}, \pi^{t+1}_{S_{ij}}). \tag{6.32}$$

The global propagation algorithm is similar to Algorithm 6.3 by using (6.30) (reps. (6.31), (6.32)) instead of (6.11) (reps. (6.12), (6.13)).

The following proposition shows that at each level of the global propagation procedure, the junction tree encodes the same joint distribution:

Proposition 6.9 *Let* $\pi^t_{\mathcal{JT}}$ *be the joint distribution relative to a junction tree* $\mathcal{JT}$ *at level* t*. Let* $\pi^{t+1}_{\mathcal{JT}}$ *be the resulted joint distribution after the modification of a cluster* C_i *using the above procedure. Then,*

$$\pi^t_{\mathcal{JT}} = \pi^{t+1}_{\mathcal{JT}}. \tag{6.33}$$

From Propositions 6.8 and 6.9 we deduce that from the initialization to the global consistency level, the junction tree encodes the same joint distribution.

Proposition 6.10 *Let π_m be the joint distribution encoded by ΠG_p (using (5.6)). Let $\pi^C_{\mathcal{JT}}$ be the joint distribution encoded by $\mathcal{JT}$ after the global propagation procedure (using (6.26)). Then,*

$$\pi_m = \pi^C_{\mathcal{JT}}. \tag{6.34}$$

The following proposition shows that the collect and distribute phases are enough to make the junction tree globally consistent.

Proposition 6.11 *The global consistency is reached after the collect and distribute phases.*

Step 3: Marginalization. Using the consistent junction tree obtained from the previous phase, we can now compute for any variable of interest A, the possibility measure $\Pi_m(A)$ as follows:

Algorithm 6.6: Marginalization

begin

Identify a cluster C_i containing A;
Compute $\Pi_m(A)$ by marginalization of $\pi^C_{C_i}$ on A: $\Pi_m(A) \leftarrow \max_{C_i \setminus A} \pi^C_{C_i}$;

end

Handling the evidence

The above algorithm can be easily extended to handle evidence so that to compute for any variable $A \in V$, the possibility degree $\Pi_m(A \wedge e)$ where e is the total evidence. This evidence will be encoded by using a *likelihood* Λ expressed by (6.16). To handle the evidence e, we should extend the propagation procedure by transforming the initialization step so that to incorporate any certain information. Indeed, we should encode the evidence e as a likelihood (using (6.16)), then, we incorporate it into the junction tree by adding these two steps to the initialization procedure:

- For any instanciated variable A, encode the observation $A = a$ as a likelihood Λ_A using (6.16).
- Identify a cluster C_i containing A: $\pi^I_{C_i} \leftarrow min(\pi^I_{C_i}, \Lambda_A)$.

Through global propagation and under the assumption that we have an evidence e, the potential of any cluster C_i encodes $\Pi_m(C_i \wedge e)$. Then, when we marginalize any cluster potential $\pi^C_{C_i}$ into a variable A (s.t $A \subseteq C_i$) using (6.28), we obtain the possibility measure of A and e:

$$\Pi_m(A \wedge e) = \max_{C_i \setminus A} \pi^C_{C_i}. \tag{6.35}$$

However, our goal is to compute $\Pi_m(A \mid e)$, this value can be easily obtained from $\Pi_m(A \wedge e)$ by applying the definition of min-based conditioning as follows:

$$\Pi_m(A \mid e) = \begin{cases} \Pi_m(A \wedge e) & \text{if } \Pi_m(A \wedge e) < \Pi_m(e) = \max_A \Pi_m(A \wedge e) \\ 1 & \text{otherwise.} \end{cases} \quad (6.36)$$

6.4 Conclusion

In this Chapter, we have proposed a possibilistic adaptation of exact probabilistic propagation algorithms for product and min based possibilistic networks. The complexity of these algorithms is the same than the probabilistic ones.

Moreover, we have shown that the initial *unrecovered* data have no effect on the propagation process in the case of min-based possibilistic networks. This result is complementary with the one obtained in Chapter 5 on the coherence problem, where we have shown that unrecovered data correspond to redundant data.

This adaptation reinforces our first conclusion about the similarity between product-based possibilistic networks and Bayesian networks and shows that this is not exactly the case for min-based possibilistic networks where the adaptation is not direct and needs some transformations, notably on initial data since we should handle joint distributions instead of conditional ones.

The proposed algorithms have been implemented in a software called Possibilistic Networks Toolbox (**PNT**) presented in Chapter 8.

Part III

New Approach in Possibilistic Propagation

Introduction Part III

The study of different adaptations of probabilistic propagation algorithms for the possibilistic framework has shown that min-based propagation can be seen differently from the classical approaches since the minimum operator has different properties from the product operator (used in both Bayesian and product-based networks) like the *idempotency* property.

Therefore, we propose, in this part, a new propagation algorithm for min-based possibilistic networks which is not a direct adaptation of classical approaches. In particular, we will avoid the transformation of the initial network into a junction tree which is known to be a hard problem [30]. This is the aim of chapter 7.

Chapter 8 provides experimental results regarding the quality of our propagation algorithm comparing with classical approaches. Moreover, it proposes a Possibilistic Networks Toolbox (**PNT**) allowing the propagation in both min-based and product-based possibilistic networks.

Chapter 7

Anytime Propagation Algorithm for Min-Based Possibilistic Networks

7.1 Introduction

In the previous Chapter we have shown that product-based possibilistic networks are very close to Bayesian networks since conditioning is defined in the same way in the two frameworks. This is not the case with min-based networks. Indeed, the minimum operator has different properties from the product operator like the *idempotency* property. Hence, we propose, in this Chapter, a new propagation algorithm for such networks which is not a direct adaptation of probabilistic propagation algorithms, as we have done in previous Chapter. In particular, we will avoid the transformation of the initial network into a junction tree which is known to be a hard problem [30].

The proposed algorithm is an anytime algorithm. It is composed of several steps, which progressively get close to exact possibility degrees (i.e. converges to exact values). The first step, consists in transforming the initial possibilistic graph into an equivalent undirected graph, called here for simplicity *moral graph*, where each node (called cluster) contains a variable from the initial graph and its parents. The clusters are quantified by local joint possibility distributions instead of the initial conditional ones. Then, several stability procedures are used in order to guarantee that joint distribution relative to any cluster is in agreement with those of its adjacent clusters.

We start by a *simple stability procedure* which ensures that any cluster agrees with each of its adjacent clusters on the distributions defined on common variables. This procedure does

not guarantee exact marginals. Thus, we propose to improve it by using a *multiple nodes stability procedure* which ensures that any cluster agrees on the distributions defined on common variables computed from 2, 3,.., n adjacent clusters. We will consider, the case where nodes are all neighbors and also the cases where nodes are restricted to parents, children and parents with children. We also consider a *best multiple nodes stability procedure* which ensures that only best instances in the distribution of each cluster agree with the best instances in the distribution computed from several of its adjacent clusters. Finally, we propose two *consistency procedures* which ensures exact marginals. The first one is based on adding some links in the moral graph while the second procedure is based on constructing best global instances.

As we will see in Chapter 8, our algorithm provides better results than the direct adaptation of junction tree algorithm (proposed in Chapter 6).

The rest of this Chapter is organized as follows, Section 7.2 introduces the notion of *α-normalized min-based possibilistic networks*. Section 7.3 introduces the basic ideas of our propagation algorithm. Section 7.4 describes the initialization procedure. Section 7.6 details the *simple stability procedure*. Section 7.7 presents the *multiple nodes stability procedure*. Section 7.8 details the *best multiple nodes stability procedure*. Section 7.9 describes the selection of stability procedures. Section 7.10 proposes the two *consistency procedures*. Finally, Section 7.11 considers the case of integrating the evidence.
Proofs of this Chapter are given in Appendix E.
Main results of this Chapter are published in [8, 10, 11].

7.2 α-normalized possibilistic networks

We first need to introduce the notion of α-normalized possibilistic networks which will be used later to represent sub-normalized possibility distributions. This notion is an extension of possibilistic networks introduced in Chapter 5.

Definition 7.1 *An* α-normalized min-based possibilistic network *over a set of variables* V*, denoted by* $\alpha\Pi G_m$*, is composed of a DAG (Directed Acyclic Graph) where nodes represent variables and arcs encode the link between the variables as follows:*

- *if* $U_A = \emptyset$ *(i.e.* A *is a root), then the a priori possibility relative to* A *should satisfy:*

$$max_a \Pi(a) = \alpha, \forall a \in D_A,$$

- *if $U_A \neq \emptyset$, then the conditional distribution of A in the context of its parents should satisfy:*

$$max_a \Pi(a \mid u_A) = \alpha, \forall a \in D_A, u_A \in D_{U_A}.$$

If A is a binary variable, then $max(\Pi(a \mid u_A), \Pi(\neg a \mid u_A)) = \alpha$.

When $\alpha = 1$, we recover classical min-based possibilistic networks introduced in Chapter 5.

Given all the a priori and conditional possibilities, the joint distribution relative to the set V, denoted by π_m, is defined , exactly as in classical min-based possibilistic networks, by the *min-based chain rule* (5.6) expressed by:

$$\pi_m(A_1, .., A_N) = \min_{i=1..N} \Pi(A_i \mid U_{A_i}).$$

An important property of α-normalized networks is expressed by the following proposition:

Proposition 7.1 *Let $\alpha\Pi G_m$ be an α-normalized min-based possibilistic network. Let π_m be the joint distribution computed from (5.6). Then, π_m is α-normalized (in the sense of Definition 1.12). Namely:*

$$h(\pi_m) = \alpha. \tag{7.1}$$

Example 7.1 *Let us consider the α-normalized min-based possibilistic network $\alpha\Pi G_m$ composed by the DAG of Figure 7.1 and the initial distributions given in Tables 7.1 and 7.2.*

Table 7.1: Initial distributions

a	$\Pi(a)$	b	a	$\Pi(b \mid a)$	c	a	$\Pi(c \mid a)$
a_1	*1*	b_1	a_1	*1*	c_1	a_1	*0.3*
a_2	*0.9*	b_1	a_2	*0*	c_1	a_2	*1*
		b_2	a_1	*0.4*	c_2	a_1	*1*
		b_2	a_2	*1*	c_2	a_2	*0.2*

Table 7.2: Initial distributions

d	b	c	$\Pi(d \mid b \wedge c)$	d	b	c	$\Pi(d \mid b \wedge c)$
d_1	b_1	c_1	*1*	d_2	b_1	c_1	*1*
d_1	b_1	c_2	*1*	d_2	b_1	c_2	*0*
d_1	b_2	c_1	*1*	d_2	b_2	c_1	*0.8*
d_1	b_2	c_2	*1*	d_2	b_2	c_2	*1*

These a priori and conditional possibilities encode the joint distribution relative to A, B, C and D using (5.6) as follows: $\forall a, b, c, d, \pi_m(a \wedge b \wedge c \wedge d) = min(\Pi(a), \Pi(b \mid a), \Pi(c \mid d), \Pi(d \mid b \wedge c)$. For instance $\pi_m(a_1 \wedge b_2 \wedge c_2 \wedge d_1) = min(1, 0.4, 1, 1) = 0.4$. Moreover we can check that $h(\pi_m) = 1$ (see the distribution π_m given in Table 7.3).

Figure 7.1: Multiply Connected DAG of Example 7.1

Next Section, proposes basic ideas of our new propagation algorithm developed for α-normalized min-based possibilistic networks.

7.3 Basic ideas

Given an α-normalized min-based possibilistic network $\alpha\Pi G_m$, our propagation algorithm provides for each variable the set of its most plausible instances. It also allows the computation for any instance of interest a, relative to a subset of variables, its possibility degree $\Pi_m(a)$ inferred from $\alpha\Pi G_m$. To compute $\Pi_m(a)$, we first define a new possibility distribution π_a from π_m as follows:

$$\pi_a(\omega) = \begin{cases} \pi_m(\omega) & \text{if } \omega[A] = a \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

Then, from π_a, it can be checked that:

$$\Pi_m(a) = h(\pi_a) = \max_{\omega} \pi_a(\omega). \tag{7.3}$$

Indeed, $\Pi_m(a)$ is the marginal distribution relative to the instance of interest a. By definition this value is computed by marginalization of the joint distribution π_m on the instance a, thus:
$\Rightarrow \Pi_m(a) = \max_\omega \pi_m(\omega)$ if $\omega[A] = a$
$= \max_\omega \pi_a(\omega)$ (using (7.2))
$= h(\pi_a)$ (using (1.12)).

Note that, in general, π_a is *sub-normalized* i.e. $h(\pi_a) < 1$.

Example 7.2 *Let us continue Example 7.1. The initial distributions encode the joint distribution π_m given in Table 7.3 using (5.6).*
*Suppose that we are interested with the value of $\Pi_m(D = d_2)$. From the joint distribution π_m, we can compute the new possibility distribution π_{d_2} using (7.2) (see Table 7.3). Computing $\Pi_m(D = d_2)$ is then immediate since it is sufficient to take the maximal value in this new sub-normalized distribution i.e. $\Pi_m(D = d_2) = h(\pi_{d_2}) = 0.8$. This value is computed from the **global** joint distribution which is not always possible especially with a great number of variables. Thus, our aim is to find the same value, but **locally**.*

Table 7.3: Joint distributions π_m et π_{d_2}

a	b	c	d	$\pi_{\mathcal{D}}$	π_{d_2}	a	b	c	d	$\pi_{\mathcal{D}}$	π_{d_2}
a_1	b_1	c_1	d_1	*0.3*	*0*	a_2	b_1	c_1	d_1	*0*	*0*
a_1	b_1	c_1	d_2	*0.3*	*0.3*	a_2	b_1	c_1	d_2	*0*	*0*
a_1	b_1	c_2	d_1	*1*	*0*	a_2	b_1	c_2	d_1	*0*	*0*
a_1	b_1	c_2	d_2	*0*	*0*	a_2	b_1	c_2	d_2	*0*	*0*
a_1	b_2	c_1	d_1	*0.3*	*0*	a_2	b_2	c_1	d_1	*0.9*	*0*
a_1	b_2	c_1	d_2	*0.3*	*0.3*	a_2	b_2	c_1	d_2	*0.8*	*0.8*
a_1	b_2	c_2	d_1	*0.4*	*0*	a_2	b_2	c_2	d_1	*0.2*	*0*
a_1	b_2	c_2	d_2	*0.4*	*0.4*	a_2	b_2	c_2	d_2	*0.2*	*0.2*

The principle of the proposed propagation method is to shift up instances with maximal degrees to the top level via a stabilization process described below. Thus, for each variable we will provide the set of its most plausible instances. If we are interested with the more particular problem of computing *the possibility degree* $\Pi_m(a)$ *of an instance of interest a relative to a subset of variables*, then we should shift up this instance to the top level. This procedure is summarized in Figure 7.2 which explains how to compute in a local manner, the possibility degree $\Pi_m(a)$.

Note that computing $\Pi_m(a)$ corresponds to possibilistic inference with no evidence. The more general problem of computing $\Pi_m(a \mid e)$, where e is the total evidence, is advocated in Section 7.11.

Basic steps of our propagation algorithm are:

- *Initialization.* Transforms the initial network into an equivalent secondary structure, called here for simplicity *moral graph*, composed of clusters of variables obtained by adding to each node its parents. Then, quantifies the moral graph using initial conditional distributions.

- *Incorporation of an instance of interest.* Incorporates the instance of interest (if any) in the initialized moral graph.

- *Simple Stability Procedure (SSP).* Ensures that any cluster agrees with each of its adjacent clusters (i.e. neighbors) on the distributions defined on common variables.

- *Multiple nodes Stability Procedure (MSP).* Ensures that any cluster agrees on the distributions defined on common variables computed from 2, 3,.., n adjacent clusters. We will consider, the case where nodes are all neighbors and also the cases where nodes are restricted to parents, children and parents with children.

- *Best Multiple nodes Stability Procedure (BMSP).* Ensures that only best instances in the distribution of each cluster agree with the best instances in the distribution computed from several of its adjacent clusters.
- *Consistency procedure.* Ensures exact marginals. Two consistency procedures are proposed, the first is based on adding some links in the moral graph while the second is based on constructing best global instances.

Figure 7.2: Propagation algorithm (with an instance of interest a)

We now detail different steps of our propagation algorithm.

7.4 Initialization

The first step in the initialization procedure is to transform the initial network into an equivalent secondary structure, called *moral graph* for simplicity of notation, and denoted by $\mathcal{MG}$.

Each node in the moral graph $\mathcal{MG}$ is called a *cluster* and it is constructed by adding to each node (variable) from the initial network its parent set. This construction way insures that for any variable A_i corresponds only one cluster in $\mathcal{MG}$ denoted by C_i.

Between any two clusters C_i and C_j with a non-empty intersection exits an edge labeled with a *separator*, denoted by S_{ij}, containing the common variables in C_i and C_j.

Once the moral graph is constructed, the initial conditional distributions are transformed into local joints in order to quantify it. Namely, for each cluster C_i of $\mathcal{MG}$, we assign a local joint distribution relative to its variables, called *potential* and denoted by $\pi^t_{C_i}$ where t corresponds to the propagation level i.e.,
- t=**I**: corresponds to the initialized potentials,
- t=**S**: corresponds to the simple stability procedure potentials,
- t=**nP** (resp. **nC**, **nPC**, **nN**): corresponds to the n-parents (resp. n-children, n-parents-children, n-neighbors) stabilized potentials (e.g. **2P** corresponds to the two-parents stabilized potentials),
- t=**n-best-P** (resp. **n-best-C**, **n-best-PC**, **n-best-N**): corresponds to the n-best-parents stabilized potentials (resp. n-best-children, n-best-parents-children, n-best-neighbors) stabilized potentials,
- t=**C**: corresponds to the consistent potentials.

We denote by c_i and s_{ij} the possible instances of the cluster C_i and the separator S_{ij}, respectively; $c_i[A]$ denotes the instance in c_i of the variable A.

For each moral graph $\mathcal{MG}$, we can associate a unique possibility distribution defined by:

Definition 7.2 *The joint distribution associated with $\mathcal{MG}$, denoted $\pi_{\mathcal{MG}}$ is expressed by:*

$$\pi_{\mathcal{MG}}(A_1, .., A_N) = \min_{i=1..N} \pi^t_{C_i}. \tag{7.4}$$

Figure 7.3: Moral graph of the DAG in Figure 7.1

The outline of this first phase of the initialization procedure is as follows:

Algorithm 7.1: From conditional to local joint distributions

begin

1. **Building the moral graph**:
- For each variable A_i, form a cluster $C_i = \{A_i\} \cup U_{A_i}$;
- Between any two clusters C_i and C_j with a non-empty intersection, add an edge labeled with a separator S_{ij} corresponding to their intersection.
2. **Quantify the moral graph**:
For each cluster C_i: $\pi^I_{C_i} \leftarrow \Pi(A_i \mid U_{A_i})$;

end

It is clear that the joint distribution encoded by the initialized moral graph is equivalent to the one encoded by the initial network since the local joint distributions in clusters are equal to the initial local conditional distributions.

Example 7.3 *Let us consider the network treated in Example 7.1. The moral graph corresponding to this network is represented in Figure 7.3. This moral graph contains four clusters (i.e. A, AB, AC, BCD) relative respectively to A, B, C and D. The initial distributions are transformed into local joint ones as shown by Table 7.4.*

Table 7.4: Initialized potentials of A, AB, AC and BCD

a	π^I_A	a	b	π^I_{AB}	a	c	π^I_{AC}	b	c	d	π^I_{BCD}	b	c	d	π^I_{BCD}
a_1	*1*	a_1	b_1	*1*	a_1	c_1	*0.3*	b_1	c_1	d_1	*1*	b_2	c_1	d_1	*1*
a_2	*0.9*	a_1	b_2	*0.4*	a_1	c_2	*1*	b_1	c_1	d_2	*1*	b_2	c_1	d_2	*0.8*
		a_2	b_1	*0*	a_2	c_1	*1*	b_1	c_2	d_1	*1*	b_2	c_2	d_1	*1*
		a_2	b_2	*1*	a_2	c_2	*0.2*	b_1	c_2	d_2	*0*	b_2	c_2	d_2	*1*

7.5 Incorporation of an instance of interest

The proposed propagation algorithm allows the computation for each variable the set of its most plausible instances. If we are interested with the more particular problem of computing the possibility degree $\Pi_m(a)$ of an instance of interest a relative to a subset of variables, then we should incorporate it in $\mathcal{MG}$ such that the possibility distribution obtained from $\mathcal{MG}$ is equal to π_a computed using (7.2). This can be obtained by modifying the potential of the

clusters relative to the instance of interest $a = a_1 \wedge, .., \wedge a_M$ i.e. $\forall i \in \{1, .., M\}$,

$$\pi^I_{C_i}(c_i) \leftarrow \begin{cases} \pi^I_{C_i}(c_i) & \text{if } c_i[A_i] = a_i \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

The following proposition shows that the moral graph obtained by incorporating the instance of interest a leads, indeed, to the possibility distribution π_a.

Proposition 7.2 *Let $\alpha\Pi G_m$ be an an α-normalized min-based possibilistic network. Let $\mathcal{MG}$ be the moral graph corresponding to $\alpha\Pi G_m$ given by the initialization procedure.*

Let π_a be the joint distribution given by (7.2) (which is obtained after incorporating the instance of interest a). Let $\pi_{\mathcal{MG}}$ be the joint distribution encoded by $\mathcal{MG}$ (given by (7.4)) after the initialization procedure. Then,

$$\pi_a = \pi^I_{\mathcal{MG}}. \tag{7.6}$$

Example 7.4 *Let us continue Example 7.1 and suppose that we are interested with the value of $\Pi_m(D = d_2)$. Table 7.5 represents the potential of the cluster BCD after incorporating this instance.*

Table 7.5: Initialized potential of BCD after incorporating the evidence $D = d_2$

b	c	d	π^I_{BCD}	b	c	d	π^I_{BCD}
b_1	c_1	d_1	*0*	b_2	c_1	d_1	*0*
b_1	c_1	d_2	*1*	b_2	c_1	d_2	*0.8*
b_1	c_2	d_1	*0*	b_2	c_2	d_1	*0*
b_1	c_2	d_2	*0*	b_2	c_2	d_2	*1*

The following subsections present several stabilizing procedures which aim to approach the exact value of $h(\pi_a)$ (hence $\Pi_m(a)$). They are based on the notion of stability, which means that adjacent clusters agree on marginal distributions defined on common variables.

7.6 Simple Stability Procedure (SSP)

Simple stability procedure ensures that any cluster agrees with each of its adjacent clusters (i.e. neighbors) on the distributions defined on common variables. More formally,

Definition 7.3 *Let C_i and C_j be two adjacent clusters in a moral graph $\mathcal{MG}$ and let S_{ij} be their separator. The separator S_{ij} is said to be **stable** if:*

$$\max_{C_i \setminus S_{ij}} \pi^t_{C_i} = \max_{C_j \setminus S_{ij}} \pi^t_{C_j} \tag{7.7}$$

where $\max_{C_i \setminus S_{ij}} \pi^t_{C_i}$ *(resp.* $\max_{C_j \setminus S_{ij}} \pi^t_{C_j}$*) is the marginal distribution of* S_{ij} *defined from* $\pi^t_{C_i}$ *(resp.* $\pi^t_{C_j}$*).*

A moral graph $\mathcal{MG}$ *is said to be* ***stable*** *if all of its separators are stable.*

The simple stability procedure is performed via a message passing mechanism between different clusters. Each separator *collects* information from its adjacent clusters, then *diffuses* it to each of them, in order to update them by taking the minimum between their initial potential and the one diffused by their separator. This operation is repeated until there is no modification on the cluster's potentials. The potentials of any adjacent clusters C_i and C_j (with separator S_{ij}) are updated as follows:

- *Collect evidence (Update separator)*:

$$\pi^{t+1}_{S_{ij}} \leftarrow min(\max_{C_i \setminus S_{ij}} \pi^t_{C_i}, \max_{C_j \setminus S_{ij}} \pi^t_{C_j}). \tag{7.8}$$

- *Distribute evidence (Update clusters)*:

$$\pi^{t+1}_{C_i} \leftarrow min(\pi^t_{C_i}, \pi^{t+1}_{S_{ij}}). \tag{7.9}$$

$$\pi^{t+1}_{C_j} \leftarrow min(\pi^t_{C_j}, \pi^{t+1}_{S_{ij}}). \tag{7.10}$$

These two steps are repeated until reaching the stability of all clusters as described by the following procedure:

Algorithm 7.2: Simple stability procedure

begin
  while $\mathcal{MG}$ *is* not stable **do**
    for *each separator* S_{ij} **do**
      - Collect evidence in S_{ij} from C_i and C_j using (7.8);
      - Distribute evidence from S_{ij} to C_i and C_j using (7.9) and (7.10);
end

In this procedure, a separator will be treated if one of its corresponding clusters has been modified. Moreover, a cluster will be treated if one of its corresponding separators has been modified. Thus, the moral graph is considered as stable if none of its clusters has been modified.

It can be shown that the simple stability is reached after a finite number of message passes, which can be evaluated as follows:
Let N be the number of clusters, M be the number of separators and P the number of values in

the possibilistic scale relative to all the clusters. The iteration in the simple stability procedure is repeated until there is no modification in the clusters. The maximal number of iterations occurs when a degree is modified in one cluster during one iteration, thus we can have at most $N * P$ iterations. Each iteration runs $O(M)$ times the Collect-Distribute evidence. Thus, the theoretical complexity is $O(N * M * P)$ and hence the stability is a *polynomial* procedure.

The following proposition shows that at each level of the simple stability procedure, the moral graph encodes the same joint distribution:

Proposition 7.3 *Let $\pi^t_{\mathcal{MG}}$ be the joint distribution relative to a moral graph $\mathcal{MG}$ at level t. Let $\pi^{t+1}_{\mathcal{MG}}$ be the resulted joint distribution after the modification of two adjacent clusters C_i and C_j using equations (7.8), (7.9) and (7.10). Then,*

$$\pi^t_{\mathcal{MG}} = \pi^{t+1}_{\mathcal{MG}}. \tag{7.11}$$

From Propositions 7.2 and 7.3 we deduce that from the initialization to the simple stability level, the moral graph encodes the same joint distribution:

Proposition 7.4 *Let π_a be the joint distribution given by (7.2). Let $\pi^S_{\mathcal{MG}}$ be the joint distribution encoded by $\mathcal{MG}$ after the simple stability procedure. Then,*

$$\pi_a = \pi^S_{\mathcal{MG}}. \tag{7.12}$$

The following proposition shows that if a moral graph is stabilized, then the maximum value of all its cluster's potentials is the same.

Proposition 7.5 *Let $\mathcal{MG}$ be a stabilized moral graph. Then, $\forall C_i$,*

$$\alpha = \max \pi^S_{C_i}. \tag{7.13}$$

Remark: Let $d = (C_1, ..., C_N)$ be any ordering of the clusters such that $U_{A_i} \subseteq \{A_1, ..A_{i-1}\}$. In the implementation proposed in Appendix F, the order in which messages circulate during the simple stability procedure depends on d. Indeed, we start with the last cluster in d which is stabilized w.r.t. of all its neighbors, then its predecessor will be treated and it will be stabilized w.r.t. of all its neighbors except those who already use it. A cycle is achieved when all the clusters are treated. This process will be repeated until reaching the stability.

Example 7.5 *Figure 7.4 illustrates one cycle of the message passing during the stabilization procedure if we consider that the clusters are ordered as follows: $d = (A, AB, AC, BCD)$:*
1-2: communication between BCD and AC via the separator C,
3-4: communication between BCD and AB via the separator B,

Figure 7.4: Message passing during the simple stability procedure

5-6: communication between AC and A via the separator A,
7-8: communication between AC and AB via the separator A,
9-10: communication between AB and A via the separator A.

Example 7.6 *Let us consider the moral graph initialized in Example 7.3. Note first that this moral graph is not stable. For instance, the separator A between the two clusters AB and A is not stable since* $\max_{AB\setminus A} \pi^I_{AB}(a_2) = 1 \neq \pi^I_A(a_2) = 0.9$.
Suppose now that we are only interested with the most plausible instances in each variable, then we will apply the simple stability procedure (arrows in Figure 7.4 illustrate one message passes during SSP). At stability level we obtain the potentials given in Table 7.6. Note that, the maximum potential is the same in the four clusters i.e. $max\pi^S_A = max\pi^S_{AB} = max\pi^S_{AC} = max\pi^S_{BCD} = 1$. *From Table 7.6, we can deduce that the most plausible instance relative to A (resp. B, C, D) is* a_1 *(resp.* b_1, c_2, d_1*).*
Suppose now that we are interested with the value of $\Pi_m(D = d_2)$*, then we should incorporate* d_2 *in the moral graph as already done in Example 7.4 and apply the simple stability procedure. The stabilized potentials are given in Table 7.7. Note also that* $max\pi^S_A = max\pi^S_{AB} = max\pi^S_{AC} = max\pi^S_{BCD} = 0.9$.

Table 7.6: Stabilized potentials

a	π^S_A	a	b	π^S_{AB}	a	c	π^S_{AC}	b	c	d	π^S_{BCD}	b	c	d	π^S_{BCD}
a_1	***1***	a_1	b_1	***1***	a_1	c_1	*0.3*	b_1	c_1	d_1	*0.9*	b_2	c_1	d_1	*0.9*
a_2	*0.9*	a_1	b_2	*0.4*	a_1	c_2	***1***	b_1	c_1	d_2	*0.9*	b_2	c_1	d_2	*0.8*
		a_2	b_1	*0*	a_2	c_1	*0.9*	b_1	c_2	d_1	***1***	b_2	c_2	d_1	*0.9*
		a_2	b_2	*0.9*	a_2	c_2	*0.2*	b_1	c_2	d_2	*0*	b_2	c_2	d_2	*0.9*

Table 7.7: Stabilized potentials with $D = d_2$

a	π^S_A	a	b	π^S_{AB}	a	c	π^S_{AC}	b	c	d	π^S_{BCD}	b	c	d	π^S_{BCD}
a_1	*0.9*	a_1	b_1	*0.9*	a_1	c_1	*0.3*	b_1	c_1	d_1	*0*	b_2	c_1	d_1	*0*
a_2	*0.9*	a_1	b_2	*0.4*	a_1	c_2	*0.9*	b_1	c_1	d_2	*0.9*	b_2	c_1	d_2	*0.8*
		a_2	b_1	*0*	a_2	c_1	*0.9*	b_1	c_2	d_1	*0*	b_2	c_2	d_1	*0*
		a_2	b_2	*0.9*	a_2	c_2	*0.2*	b_1	c_2	d_2	*0*	b_2	c_2	d_2	*0.9*

It is important to mention that the simple stability procedure does not guarantee that the degree α corresponds to the exact degree $\Pi_m(a) = h(\pi_a)$ since the equality $h(\pi^S_{MG}) = \alpha$ is not always verified. More formally:

Proposition 7.6 *Let α be the maximal degree generated by the simple stability procedure (which is unique c.f. Proposition 7.5). Then,*

$$\alpha \geq \Pi_m(a). \tag{7.14}$$

Indeed, from 7.2 and 7.3, we have,
$\Pi_m(a) = h(\pi_a) = \max_\omega \pi_a(\omega) = \max_\omega \{min_{i=1..N} \pi^S_{C_i}(\omega),\ \text{s.t. } \omega[A] = a\}$.
Moreover, $\forall C_i, \pi^S_{C_i}(\omega) \leq \alpha$ (since α is the maximum value in all clusters), thus $\Pi_m(a) \leq \alpha$.

For instance, we can check in the previous example that $h(\pi^S_{\mathcal{MG}}) = 0.8 > 0.9$. Nevertheless, as we will see in next Chapter, experimental results show that, in general, this equality holds.

7.7 Multiple nodes Stability Procedure (MSP)

As mentioned before, the simple stability procedure does not always guarantee exact marginals. Thus, our idea is to improve it by considering stability with respect to a greater number of adjacent clusters. Indeed, for each cluster C_i, corresponding to a node A_i, we can distinguish these particular sets within its adjacent clusters:
- *parents*, containing the clusters relative to the parents of A_i,
- *children*, containing the clusters relative to the children of A_i,
- *parents-children*, containing parents of A_i and its children,
- *neighbors*, containing all adjacent clusters of A_i.

Example 7.7 *Let us consider the network composed of the DAG in Figure 7.5 (a) and the initial distributions given by Tables 7.8, 7.9, 7.10, 7.11, 7.12 and 7.13. Let us consider, for instance, the cluster ABC (see the moral graph in Figure 7.5 (b)), then its* parents *are A and B, it* children *are ABE and CFG, its* parents-children *are A, B, ABE and CFG and its* neighbors *are A, B, ABE, CFG and ABD. This example will be continued to explain multiple nodes stability procedure.*

Figure 7.5: DAG and moral graph of Example 7.7

Table 7.8: A priori distributions

a	$\Pi(a)$	b	$\Pi(b)$
a_1	*1*	b_1	*1*
a_2	*0.9363*	b_2	*0.8552*
a_3	*0.2240*		

Table 7.9: Conditional distribution of C in the context of A and B

a	b	c	$\Pi(c \mid a \wedge b)$	a	b	c	$\Pi(c \mid a \wedge b)$	a	b	c	$\Pi(c \mid a \wedge b)$
a_1	b_1	c_1	*0.5463*	a_2	b_1	c_1	*1*	a_3	b_1	c_1	*0.5692*
a_1	b_1	c_2	*1*	a_2	b_1	c_2	*0.5088*	a_3	b_1	c_2	*1*
a_1	b_1	c_3	*0.6707*	a_2	b_1	c_3	*0.5064*	a_3	b_1	c_3	*0.5590*
a_1	b_2	c_1	*1*	a_2	b_2	c_1	*0.1795*	a_3	b_2	c_1	*1*
a_1	b_2	c_2	*0*	a_2	b_2	c_2	*1*	a_3	b_2	c_2	*0.2239*
a_1	b_2	c_3	*0.7141*	a_2	b_2	c_3	*0.5655*	a_3	b_2	c_3	*0.4059*

Table 7.10: Conditional distribution of D in the context of A and B

a	b	d	$\Pi(d \mid a \wedge b)$	a	b	c	$\Pi(d \mid a \wedge b)$	a	b	d	$\Pi(d \mid a \wedge b)$
a_1	b_1	d_1	*0.8215*	a_2	b_1	c_1	*1*	a_3	b_1	c_1	*0.3328*
a_1	b_1	d_2	*1*	a_2	b_1	c_2	*0.9946*	a_3	b_1	c_2	*1*
a_1	b_2	d_1	*1*	a_2	b_2	c_1	*0.7363*	a_3	b_2	c_1	*1*
a_1	b_2	d_2	*0*	a_2	b_2	c_2	*1*	a_3	b_2	c_2	*0.2788*

Ideally, we want to perform the n-neighbors stability where n is the cardinality of the neighbor set relative to each cluster. In other terms, each cluster will be stabilized with respect to all its neighbors. However, this can be impossible especially when many clusters share the same variables. To avoid this problem, we propose several stability procedures where we consider a less number of adjacent clusters. More precisely, we study four cases (where $n > 1$):

- *n-parents* stability ensuring for each cluster its stability w.r.t its parents,
- *n-children* stability ensuring for each cluster its stability w.r.t. its children,
- *n-parents-children* stability ensuring for each cluster its stability w.r.t its parents and children,
- *n-neighbors* stability ensuring for each cluster its stability w.r.t. all its adjacent clusters (i.e. neighbors).

In *n-parents* (resp. *n-children*, *n-parents-children*, *n-neighbors*) stability procedure, we will vary the number of parents (resp. children, parents-children, neighbors) by first considering

Table 7.11: Conditional distribution of E in the context of A, B and C

a	b	c	e	$\Pi(e \mid a \wedge b \wedge c)$	a	b	c	e	$\Pi(e \mid a \wedge b \wedge c)$	a	b	c	e	$\Pi(e \mid a \wedge b \wedge c)$
a_1	b_1	c_1	e_1	*0.7563*	a_2	b_1	c_1	e_1	*1*	a_3	b_1	c_1	e_1	*0.9959*
a_1	b_1	c_1	e_2	*1*	a_2	b_1	c_1	e_2	*0.7605*	a_3	b_1	c_1	e_2	*1*
a_1	b_1	c_1	e_3	*0.4750*	a_2	b_1	c_1	e_3	*0.2092*	a_3	b_1	c_1	e_3	*0.2387*
a_1	b_1	c_2	e_1	*0.8161*	a_2	b_1	c_2	e_1	*1*	a_3	b_1	c_2	e_1	*0.8246*
a_1	b_1	c_2	e_2	*1*	a_2	b_1	c_2	e_2	*0.6098*	a_3	b_1	c_2	e_2	*1*
a_1	b_1	c_2	e_3	*0*	a_2	b_1	c_2	e_3	*0.1911*	a_3	b_1	c_2	e_3	*0.2965*
a_1	b_1	c_3	e_1	*0.7544*	a_2	b_1	c_3	e_1	*1*	a_3	b_1	c_3	e_1	*0.3467*
a_1	b_1	c_3	e_2	*1*	a_2	b_1	c_3	e_2	*0.6776*	a_3	b_1	c_3	e_2	*1*
a_1	b_1	c_3	e_3	*0.6902*	a_2	b_1	c_3	e_3	*0*	a_3	b_1	c_3	e_3	*0.4550*
a_1	b_2	c_1	e_1	*1*	a_2	b_2	c_1	e_1	*0.5346*	a_3	b_2	c_1	e_1	*1*
a_1	b_2	c_1	e_2	*0.4855*	a_2	b_2	c_1	e_2	*1*	a_3	b_2	c_1	e_2	*0.9249*
a_1	b_2	c_1	e_3	*0.3390*	a_2	b_2	c_1	e_3	*0.1979*	a_3	b_2	c_1	e_3	*0.4534*
a_1	b_2	c_2	e_1	*1*	a_2	b_2	c_2	e_1	*0.7398*	a_3	b_2	c_2	e_1	*1*
a_1	b_2	c_2	e_2	*0*	a_2	b_2	c_2	e_2	*1*	a_3	b_2	c_2	e_2	*0.5185*
a_1	b_2	c_2	e_3	*0*	a_2	b_2	c_2	e_3	*0.4158*	a_3	b_2	c_2	e_3	*0.2562*
a_1	b_2	c_3	e_1	*1*	a_2	b_2	c_3	e_1	*0.6012*	a_3	b_2	c_3	e_1	*1*
a_1	b_2	c_3	e_2	*0.3983*	a_2	b_2	c_3	e_2	*1*	a_3	b_2	c_3	e_2	*0*
a_1	b_2	c_3	e_3	*0.7246*	a_2	b_2	c_3	e_3	*0.4443*	a_3	b_2	c_3	e_3	*0.4154*

Table 7.12: Conditional distribution of F in the context of E

e	f	$\Pi(f \mid e)$	e	f	$\Pi(f \mid e)$	e	f	$\Pi(f \mid e)$
e_1	f_1	*0.7646*	e_2	f_1	*1*	e_3	f_1	*0.5797*
e_1	f_2	*1*	e_2	f_2	*0.6381*	e_3	f_2	*1*

two parents (resp. children, parents-children, neighbors), then three parents (resp. children, parents-children, neighbors) until reaching n parents (resp. children, parents-children, neighbors) where n is the cardinality of parents (resp. children, parents-children, neighbors) relative to each cluster.

To illustrate the multiple nodes stability, we only present **two-parents stability**. The principle of this procedure is to ensure for each cluster, having at least two parents, its stability with respect to each pair of them. More formally:

Definition 7.4 *Let C_i be a cluster in a moral graph $\mathcal{MG}$, let C_j and C_k be two parents of C_i. Let S_{ij} be the separator between C_i and C_j and S_{ik} be the separator between C_i and C_k. Let $C = C_j \cup C_k$ and let $S = S_{ij} \cup S_{ik}$. Let π_C be the joint distribution computed from $\pi^t_{C_j}$ and $\pi^t_{C_k}$. The cluster C_i is said to be stable with respect to its two parents C_j and C_k if:*

$$\max_{C_i \setminus S} \pi^t_{C_i} = \max_{C \setminus S} \pi_C, \tag{7.15}$$

Table 7.13: Conditional distribution of C in the context of A and B

c	f	g	$\Pi(g \mid c \wedge f)$	c	f	g	$\Pi(g \mid c \wedge f)$	c	f	g	$\Pi(g \mid c \wedge f)$
c_1	f_1	g_1	*0.2742*	c_2	f_1	g_1	*1*	c_3	f_1	g_1	*0.2126*
c_1	f_1	g_2	*1*	c_2	f_1	g_2	*0.5067*	c_3	f_1	g_2	*1*
c_1	f_1	g_3	*0.4066*	c_2	f_1	g_3	*0.2142*	c_3	f_1	g_3	*0.6568*
c_1	f_2	g_1	*1*	c_2	f_2	g_1	*0*	c_3	f_2	g_1	*1*
c_1	f_2	g_2	*0.2104*	c_2	f_2	g_2	*1*	c_3	f_2	g_2	*0.4538*
c_1	f_2	g_3	*0.3430*	c_2	f_2	g_3	*0.8111*	c_3	f_2	g_3	*0.6424*

where $\max_{C_i \setminus S} \pi^t_{C_i}$ *(resp.* $\max_{C \setminus S} \pi_C$*) is the marginal distribution of* S *defined from* $\pi^t_{C_i}$ *(resp.* π_C*).*

In a similar way, a cluster C_i *is said to be* ***two-parents stable*** *if it is stable with respect to each pair of its parents. Then, a moral graph* $\mathcal{MG}$ *is said to be* ***two-parents stable*** *if all of its clusters are two-parents stable.*

The updating of any cluster C_i with respect to two of its parents C_j and C_k is performed as follows:

- *Compute the potential of* C *using* C_j *and* C_k:

$$\pi_C \leftarrow min(\pi^t_{C_j}, \pi^t_{C_k}). \tag{7.16}$$

- *Compute the potential of* S *using* C:

$$\pi_S \leftarrow \max_{C \setminus S} \pi_C. \tag{7.17}$$

- *Update the potential of* C_i *using* S:

$$\pi^{t+1}_{C_i} \leftarrow min(\pi^t_{C_i}, \pi_S). \tag{7.18}$$

Algorithm 7.3: Two-parents stability procedure

begin
  Modification ← true;
  while Modification **do**
    Stabilize $\mathcal{MG}$ using the simple stability procedure;
    Modification ← false;
    i ← 1;
    while *i <= N and not Modification* **do**
      Stabilize C_i with respect to each pair of its parents, C_j and C_k using (7.18);
      if C_i *potential changes* **then** *Modification*← true;
      i ← i+1;
end

Note that the simple stability procedure is used after each modification since it is an efficient procedure which allows to minimize the application of the two-parents stability.

The following proposition shows that at each level of the two-parents stability procedure, the moral graph encodes the same joint distribution:

Proposition 7.7 *Let $\pi^t_{\mathcal{MG}}$ be the joint distribution relative to a moral graph $\mathcal{MG}$ at level t. Let $\pi^{t+1}_{\mathcal{MG}}$ be the resulted joint distribution after the modification of a cluster C_i with respect to its two parents C_j and C_k using equation (7.18). Then,*

$$\pi^t_{\mathcal{MG}} = \pi^{t+1}_{\mathcal{MG}}. \tag{7.19}$$

From Propositions 7.2, 7.4 and 7.7 we deduce that from the initialization to the two-parents stability level, the moral graph encodes the same joint distribution:

Proposition 7.8 *Let π_a be the joint distribution given by (7.2). Let $\pi^{2P}_{\mathcal{MG}}$ be the joint distribution encoded by $\mathcal{MG}$ after the two-parents stability procedure. Then,*

$$\pi_a = \pi^{2P}_{\mathcal{MG}}. \tag{7.20}$$

Example 7.8 *Let us consider the inconsistent stabilized moral graph of Example 7.6. The two-parents stabilized potential of the cluster BCD with respect to its two parents AB and AC is given by Table 7.14. Note, for instance, that the potential of $c_2 \wedge b_2 \wedge d_2$ decreases from* 0.9 *to* 0.4. *Thus, we should re-stabilize the moral graph using the simple stability procedure (see Table 7.15). We can check that the resulted moral graph is two-parents stabilized. Moreover, we have $h(\pi^{2P}_{\mathcal{MG}}) = 0.8$, in other terms, we have reached the consistency degree of π_{d_2}.*

Table 7.14: Two-parents stabilized potential of BCD

b	c	d	π^{2P}_{BCD}	b	c	d	π^{2P}_{BCD}
b_1	c_1	d_1	*0*	b_2	c_1	d_1	*0*
b_1	c_1	d_2	*0.3*	b_2	c_1	d_2	*0.8*
b_1	c_2	d_1	*0*	b_2	c_2	d_1	*0*
b_1	c_2	d_2	*0*	b_2	c_2	d_2	***0.4***

Table 7.15: re-stabilized potentials

a	π^S_A	a	b	π^S_{AB}	a	c	π^S_{AC}	b	c	d	π^S_{BCD}	b	c	d	π^S_{BCD}
a_1	*0.4*	a_1	b_1	*0.3*	a_1	c_1	*0.3*	b_1	c_1	d_1	*0*	b_2	c_1	d_1	*0*
a_2	*0.8*	a_1	b_2	*0.4*	a_1	c_2	*0.4*	b_1	c_1	d_2	*0.3*	b_2	c_1	d_2	*0.8*
		a_2	b_1	*0*	a_2	c_1	*0.8*	b_1	c_2	d_1	*0*	b_2	c_2	d_1	*0*
		a_2	b_2	*0.8*	a_2	c_2	*0.2*	b_1	c_2	d_2	*0*	b_2	c_2	d_2	*0.4*

The following proposition shows that the two-parents stability improves the simple stability procedure:

Proposition 7.9 *Let α_1 be the maximal degree generated by the simple stability procedure (which is unique c.f. Proposition 7.5). Let α_2 be the maximal degree generated by the two-parents stability. Then,*

$$\alpha_1 \geq \alpha_2 \geq \Pi_m(a). \tag{7.21}$$

The proof of this proposition is immediate since before applying the two-parents stability procedure we use the simple stability procedure, then (7.18) will be applied on each cluster w.r.t. each pair of its parents. Thus, the potential of each cluster will be decreased or saved since from (7.18) $\pi_{C_i}^{t+1} = min(\pi_{C_i}^t, \pi_S)$ and hence the maximal potential relative to each cluster (i.e. α_1) will be decreased or saved.

Example 7.9 *Let us continue Example 7.7. Let the instance of interest be $b_1 \wedge d_1 \wedge g_2$. The two-parents stabilized potentials are given by Tables 7.16, 7.17, 7.18, 7.19, 7.20 and 7.21. We can check that these potentials are inconsistent since $\alpha_2 = 0.8215$ while $\Pi_m(b_1 \wedge d_1 \wedge g_2) = 0.8161$. Moreover, it can be checked the exact value of $\Pi_m(b_1 \wedge d_1 \wedge g_2)$ is not reached neither with n-parents nor with n-children stability. However, it can be reached with two-parents-children stability.*

Table 7.16: Two-parents stabilized potentials of A and B

a	π_A^{2P}	b	π_B^{2P}
a_1	*0.8215*	b_1	*0.8215*
a_2	*0.8215*	b_2	*0*
a_3	*0.2240*		

Table 7.17: Two-parents stabilized potentials of ABC

a	b	c	π_{ABC}^{2P}	a	b	c	π_{ABC}^{2P}	a	b	c	π_{ABC}^{2P}
a_1	b_1	c_1	*0.5463*	a_2	b_1	c_1	*0.8215*	a_3	b_1	c_1	*0.2240*
a_1	b_1	c_2	*0.8215*	a_2	b_1	c_2	*0.5088*	a_3	b_1	c_2	*0.2240*
a_1	b_1	c_3	*0.6707*	a_2	b_1	c_3	*0.5064*	a_3	b_1	c_3	*0.2240*
a_1	b_2	-	*0*	a_2	b_2	-	*0*	a_3	b_2	-	*0*

Table 7.18: Two-parents stabilized potentials of ABD

a	b	d	π_{ABD}^{2P}	a	b	d	π_{ABD}^{2P}	a	b	d	π_{ABD}^{2P}
a_1	b_1	d_1	*0.8215*	a_2	b_1	d_1	*0.8215*	a_3	b_1	d_1	*0.2240*
a_1	b_2	-	*0*	a_2	b_2	-	*0*	-	-	-	*0*

Table 7.19: Two-parents stabilized potentials of ABCE

a	b	c	e	π^{2P}_{ABCE}	a	b	c	e	π^{2P}_{ABCE}	a	b	c	e	π^{2P}_{ABCE}
a_1	b_1	c_1	e_1	*0.5463*	a_2	b_1	c_1	e_1	*0.8215*	a_3	b_1	c_1	e_1	*0.2240*
a_1	b_1	c_1	e_2	*0.5463*	a_2	b_1	c_1	e_2	*0.7605*	a_3	b_1	c_1	e_2	*0.2240*
a_1	b_1	c_1	e_3	*0.4750*	a_2	b_1	c_1	e_3	*0.2092*	a_3	b_1	c_1	e_3	*0.2240*
a_1	b_1	c_2	e_1	*0.8161*	a_2	b_1	c_2	e_1	*0.5088*	a_3	b_1	c_2	e_1	*0.2240*
a_1	b_1	c_2	e_2	*0.8215*	a_2	b_1	c_2	e_2	*0.5088*	a_3	b_1	c_2	e_2	*0.2240*
a_1	b_1	c_2	e_3	*0*	a_2	b_1	c_2	e_3	*0.1911*	a_3	b_1	c_2	e_3	*0.2240*
a_1	b_1	c_3	e_1	*0.6707*	a_2	b_1	c_3	e_1	*0.5064*	a_3	b_1	c_3	e_1	*0.2240*
a_1	b_1	c_3	e_2	*0.6707*	a_2	b_1	c_3	e_2	*0.5064*	a_3	b_1	c_3	e_2	*0.2240*
a_1	b_1	c_3	e_3	*0.6707*	a_2	b_1	c_3	e_3	*0*	a_3	b_1	c_3	e_3	*0.2240*
a_1	b_2	-	-	*0*	a_2	b_2	-	-	*0*	a_3	b_2	-	-	*0*

Table 7.20: Two-parents stabilized potentials of EF

e	f	π^{2P}_{EF}	e	f	π^{2P}_{EF}	e	f	π^{2P}_{EF}
e_1	f_1	*0.7646*	e_2	f_1	*0.8215*	e_3	f_1	*0.5797*
e_1	f_2	*0.8215*	e_2	f_2	*0.6381*	e_3	f_2	*0.6707*

Remark: All results presented with two-parents stability are also available for any multiple nodes stability procedure.

7.8 Best Multiple nodes Stability Procedure (BMSP)

The stability procedures MSP, presented above, can be limited by the number of considered adjacent clusters. For instance, if we want to use n-neighbors stability, we can be limited in the computation of the cartesian product relative to the n-neighbors of some clusters especially when they have a high number of neighbors. In order to avoid this problem, we will relax the multiple nodes stability procedure by only computing the best instances in the cartesian product, denoted by *best_ nodes_ instances*.

The main motivation, of this procedure called, *best multiple nodes stability*, is that we only need to compute the maximal value in π_a, (i.e. $h(\pi_a)$), and not the whole distribution π_a. The idea is to first apply the simple stability procedure, then to cover for any cluster C_i its n considered adjacent clusters (which can be parents, children, parents-children, neighbors) in order to construct the best instances relative to these clusters.

Once the elements in *best_ nodes_ instances* are constructed, we can compute the best instances relative to the separators existing between C_i and its parents (resp. children, parents-children, neighbors) (denoted by *sep_ instances_ from_ nodes*) and compare them with the

Table 7.21: Two-parents stabilized potentials of CFG

c	f	g	π^{2P}_{CFG}	c	f	g	π^{2P}_{CFG}	c	f	g	π^{2P}_{CFG}
c_1	f_1	g_1	*0*	c_2	f_1	g_1	*0*	c_3	f_1	g_1	*0*
c_1	f_1	g_2	*0.8215*	c_2	f_1	g_2	*0.5067*	c_3	f_1	g_2	*0.6707*
c_1	f_1	g_3	*0*	c_2	f_1	g_3	*0*	c_3	f_1	g_3	*0*
c_1	f_2	g_1	*0*	c_2	f_2	g_1	*0*	c_3	f_2	g_1	*0*
c_1	f_2	g_2	*0.2104*	c_2	f_2	g_2	*0.8215*	c_3	f_2	g_2	*0.4538*
c_1	f_2	g_3	*0*	c_2	f_2	g_3	*0*	c_3	f_2	g_3	*0*

ones computed from C_i (denoted by *sep_ instances_ from_ cluster*). These two sets are incoherent if instances in *sep_ instances_ from_ nodes* are different from those in *sep_ instances_ from_ cluster*. In this case, we should decrease the potential of incoherent instances in C_i (i.e. incoherent with *sep_ instances_ from_ nodes*) to the next value in the possibilistic scale containing all the degrees relative to the considered adjacent clusters and re-stabilize the moral graph again.

The following algorithm summarizes the outline of *best multiple nodes stability procedure* in the case of parents[1].

Algorithm 7.4: n-best parents stability procedure

begin
- *Modification* ← true;
- **while** Modification **do**
  - Stabilize $\mathcal{MG}$ using the simple stability procedure;
  - Let α be the maximum (best) degree in clusters;
  - *Modification* ← false;
  - i ← 1;
  - **while** *i <= N and not Modification* **do**
    - Let *parents* be the parents of C_i;
    - Let *separators* be the set of separators existing between C_i and *parents*;
    - Let *best_ nodes_ instances* be the set of best instances in the joint distribution relative to *parents*;
    - Let *sep_ instances_ from_ nodes* be the instances relative to *separators* computed from *best_ nodes_ instances*;
    - Let *sep_ instances_ from_ cluster* be the instances relative to *separators* computed from C_i;
    - **if** sep_instances_from_nodes ≠ sep_instances_from_cluster **then**
      - Decrease the potential of incoherent instances in C_i to the next value in the possibilistic scale relative to *parents*;
      - *Modification*←true;
  - i ← i+1;

end

[1]This algorithm is the same for children (resp. parents-children, neighbors) by simply replacing parents by children (resp. parents-children, neighbors).

Remark: In the proposed implementation detailed in Appendix F, we use the following heuristics to determine the order in which clusters in *parents* are treated to compute *best_ nodes_ instances*:

- we start by the clusters having the *less* number of best instances,
- when two or more clusters have the same number of best instances, we choose the candidate cluster with the *large* number of variables appearing in the treated cluster.

Another possible heuristic, will be to choose the cluster adding the less number of instances in *best_ nodes_ instances*.

The following example illustrates *n-best-neighbors* stability procedure:

Example 7.10 *Let us consider the cluster $CEFG$ in Figure 7.6, having three neighbors ABC, CDE and F. The Figure shows the best instances in each cluster (for instance the best instance in the cluster F is f_1). The best instances in the joint distribution relative to the nodes in ABC, CDE and F (i.e. best_ nodes_ instances) is constructed by first considering the cluster F since it contains the less number of best instances then, CDE and finally ABC.*
From best_ nodes_ instances, we can check that the best instances relative to the three separators C, E and F are sep_ instances_ from_ nodes=$\{c_1 \wedge e_1 \wedge f_1, c_1 \wedge e_2 \wedge f_1\}$. However, from $CEFG$, we have sep_ instances_ from_ cluster=$\{c_1 \wedge e_1 \wedge f_1, c_2 \wedge e_1 \wedge f_2\}$. It is clear that the instance $c_2 \wedge e_1 \wedge f_2 \wedge g_1$ is incoherent with sep_ instances_ from_ nodes, thus we should decrease its degree from α to the next degree in the possibilistic scale relative to ABC, CDE and F and re-stabilize the moral graph using the simple stability procedure.

Figure 7.6: Example of n-best-neighbors stability

7.9 Selection of stability procedures

This section summarizes how to use the stability procedures presented in the above section. Indeed, ideally we want to apply n-neighbor stability. However, as shown by experimental results in next Chapter, this procedure is limited by its running time. Moreover, it can be limited by the size of joint distributions it computed. Thus, the idea is to choose the appropriate stability procedure regarding to the allowed running time and the system capacity. This

choice is explained by a *global propagation algorithm*.

Figure 7.7 summarizes the preference between multiple nodes stability procedures (MSP) and best multiple nodes stability procedures (BMSP) starting by those providing best approximation of exact marginals (see experimental results in next Chapter). Note that *n-parents* (resp. *n-best-parents*) and *n-children* (resp. *n-best-children*) stability procedures are incomparable[2] but they provide the same approximation of exact marginals (as it will be shown by Table 8.2). However, *n-parents* (resp. *n-best-parents*) stability is more reasonable with respect to the running time that is why, it selected before *n-children* (resp. *n-best-children*) stability.

Figure 7.7: Preference between stability procedure

The selection of stability procedure can be described as follows:
We first run *n-neighbors* stability (where n is the number of neighbors relative to each cluster). If this procedure is not limited (either by running time or by system capacity), then we can stop the program since it is the more efficient one (as shown by experimental results in next Chapter) and the application of the remaining procedures will not improve the result.

If we cannot run *n-neighbors* stability, then we should select the more efficient stability procedure within MSP (except *n-neighbors*) respecting the running time and the system capacity. Moreover, we should select the more efficient stability procedure within BMSP respecting the running time and the system capacity. The selected procedure in BMSP should be incomparable with the procedure chosen in MSP. For instance, if we run *n-parents-children* procedure then it is useless to run the *n-best-parents-children* procedure and it is sufficient to apply *n-best-neighbors* procedure (which is incomparable with *n-parents-children*).

The following algorithm summarizes the global propagation procedure under the assumption that the running time is unlimited.

[2] *n-parents* (resp. *n-best-parents*) stability does not imply *n-children* (resp. *n-best-children*) stability and vice versa.

Algorithm 7.5: Global propagation procedure

begin
  Run n-neighbors stability procedure;
  if *blocked* **then**
    nodes_type=[neighbors, parent-children, parents, children];
    1. Selection of MSP (*the most preferred is n-1 neighbors the least preferred is 2-children*)
    MSP_chosen ← false;
    i ← 1;
    nodes_number ← n-1;
    while *(not MSP_chosen) and (i ≤ 4)* **do**
      MSP_nodes_type ← nodes_type[i];
      Run the multiple nodes stability procedure using MSP_nodes_type and nodes_number;
      if *the selected procedure is blocked* **then**
        if *nodes_number > 2* **then** nodes_number ← nodes_number-1;
        else
          i ← i+1 ;
          nodes_number ← n;
      else MSP_chosen ← true;
    2. Selection of BMSP (*the most preferred is n-best-neighbors the least preferred is n-best-children*)
    BMSP_chosen ← false;
    j ← 1;
    while *(not BMSP_chosen) and (j < i) and (j ≤ 4)* **do**
      BMSP_nodes_type ← nodes_type[j];
      Run the best multiple nodes stability procedure using BMSP_nodes_type;
      if *the selected procedure is blocked* **then** j ← j+1 ;
      else BMSP_chosen ← true;
end

If we consider the running time parameter, then we should stop the *global stability procedure* when reaching the allowed time.

7.10 Consistency procedure

Given a stabilized moral graph $\mathcal{MG}$, such that α is the maximum potential in its clusters (which is the same due to Proposition 7.5), our aim is to check if $\mathcal{MG}$ provides exact marginals (i.e. $\alpha = h(\pi^t_{\mathcal{MG}})$) by avoiding the computation of the global joint distribution which is practically impossible especially when handling a great number of variables. In what follows, we propose two ways to test and ensure consistency, the first one is based on adding some links in the moral graph while the second is based on constructing best global instances.

7.10.1 Consistency procedure by adding links

To explain the consistency procedure by adding links, we first need a further definition:

Definition 7.5 *A cluster C_i relative to the variable A_i is said to be **consistent** if for any instance u_{A_i} of U_{A_i}:*

$$\max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \alpha. \tag{7.22}$$

We provide now a practical way to check the consistency of a moral graph.

Proposition 7.10 *A moral graph $\mathcal{MG}$ is said to be **consistent** if all its clusters are consistent.*

The proof of this proposition is based on Proposition 7.1 and the following technical lemma:

Lemma 7.1 *Let $\mathcal{MG}$ be a stabilized moral graph and let $\pi^S_{\mathcal{MG}}$ be its joint distribution. If all the clusters of $\mathcal{MG}$ are consistent, then there exists an α-DAG $\mathcal{G}'$ such that its joint distribution π'_m is equal to $\pi^S_{\mathcal{MG}}$ i.e.*

$$\pi^S_{\mathcal{MG}} = \pi'_m. \tag{7.23}$$

Case of consistency. In the case where the moral graph is consistent, the computation of α is immediate with the help of Proposition 7.1 and Lemma 7.1. Indeed, it is sufficient to take α as the maximal potential of any cluster i.e. $\alpha = \pi^C_{C_i}$. Thus, if we have already incorporated an instance of interest a, then we can consider α as the exact value of $\Pi_m(a)$ i.e. $\Pi_m(a) = max\pi^C_{C_i} = \alpha$.

Case of inconsistency. If there exists a variable $A_i \in V$ where $\exists u_{A_i}$ s.t. $\max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \beta < \alpha$, then the moral graph is not yet consistent (due to Definition 7.22). In this case, the inconsistency can be dropped from each inconsistent cluster C_i by replacing for any instance u_{A_i} s.t. $\max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \beta < \alpha$, the potential β by α. However, the degree β should be retrieved. Thus, the idea is to check if the parents of A_i are linked (i.e. there exists a cluster which contains U_{A_i}) or if the degree β already exists in the parents of C_i. If none if these two cases occur, we should create new links between variables in U_{A_i}. More precisely, we select a parents of A_i and we add to its parent set the remaining variables in U_{A_i}. The quantification of the new links allows the incorporation of the degree β. The outline of this algorithm is summarized as follows:

Algorithm 7.6: Consistency procedure by adding links

begin
  Modification ← true;
  while Modification **do**
    Stabilize $\mathcal{MG}$ using the simple stability procedure;
    Let α be the maximum (best) degree in clusters;
    Modification ← false;
    i ← 1;
    while *i <= N and not Modification* **do**
      if C_i *has more than one parent* **then**
        if C_i *is inconsistent in the sense of Definition 7.5* **then**
          Let X be the set of all instances u_{A_i} s.t. $\max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \beta < \alpha$;
          if *∄ a cluster containing* U_{A_i} *and the degree* β *does not exist in the parents of* C_i **then**
            Modification ← true;
            1. **Drop inconsistency**
            For any instance u_{A_i} in X, replace the degree β by α;
            2. **Retrieve the degree** β
            2.0. *Add new links between parents*:
            Let A_j be any of the parents of A_i:
            - Add to C_j, $T = U_{A_i} \setminus \{A_j\}$: $U_{A_j} \leftarrow U_{A_j} \cup T$
            - Update the separators associated with C_j
            2.1. *Update* C_j *potential:*
            $\pi^{t+1}_{C_j} \leftarrow min(\pi^t_{C_j}, \pi_{new})$
            where $\pi_{new}(u_{A_i}) = \begin{cases} \beta & \text{if } u_{A_i} \in X \\ \alpha & \text{otherwise} \end{cases}$
      i ← i+1
end

Note that if C_i is an inconsistent cluster, then it is useless to replace α by β if there exists a cluster which contains U_{A_i} or if the degree β already exists in the parents of C_i. Indeed, this modification will not be followed by an application of the simple stability procedure (since we will not add new links) and this can give wrong results regarding the most plausible instances in each variable.

Proposition 7.11 shows that at each level of the checking and recovering consistency procedure, the moral graph encodes the same joint distribution:

Proposition 7.11 *Let* $\pi^t_{\mathcal{MG}}$ *be the joint distribution relative to a moral graph* $\mathcal{MG}$ *at level* t.

Figure 7.8: Modified Moral Graph

Let $\pi^{t+1}_{\mathcal{MG}}$ *be the resulted joint distribution obtained as result of the procedure above. Then,*

$$\pi^{t}_{\mathcal{MG}} = \pi^{t+1}_{\mathcal{MG}}. \tag{7.24}$$

From Propositions 7.2, 7.4 and 7.11 we deduce that from the initialization to the consistency level, the moral graph encodes the same joint distribution.

Proposition 7.12 *Let* π_a *be the joint distribution given by (7.2). Let* $\pi^{C}_{\mathcal{MG}}$ *be the joint distribution encoded by* $\mathcal{MG}$ *after the checking and recovering consistency procedure. Then,*

$$\pi_a = \pi^{C}_{\mathcal{MG}}. \tag{7.25}$$

Example 7.11 *Let us consider the moral graph stabilized in Example 7.6. We can check that this moral graph is inconsistent, for instance* $max\pi^{S}_{AB} = 0.9$ *while* $h(\pi_{d_2}) = 0.8$. *This is due to the cluster BCD corresponding to the variable D since*
$max(\pi^{S}_{BCD}(b_2 \wedge c_1 \wedge d_1), \pi^{S}_{BCD}(b_2 \wedge c_1 \wedge d_2)) = 0.8 < 0.9$ *and*
$max(\pi^{S}_{BCD}(b_1 \wedge c_2 \wedge d_1), \pi^{S}_{BCD}(b_1 \wedge c_2 \wedge d_2)) = 0 < 0.9$.
This means that $X = \{b_1 \wedge c_2, b_2 \wedge c_1\}$.
Thus we should modify the potential of BCD and modify for instance the cluster AC by considering B as a new parent of C (so that to respect the topological order of variable which is $d = (A, B, C, D)$ *i.e. ancestors before descendants). This entails a modification of the moral graph as shown in Figure 7.8. The new potentials of BCD and ABC are given in Table 7.22 and the re-stabilization potentials are given in Table 7.23. These potentials does not satisfy the consistency in the sense of Definition 7.5. For instance,* $max(\pi^{S}_{AB}(a_1 \wedge b_1), \pi^{S}_{AB}(a_1 \wedge b_2)) = 0.4 \neq 0.8$. *However, the parents of all clusters having more than one parent are linked, thus no modification will be made on the moral graph and the consistency procedure stops which means that the consistency is reached. Thus, the possibility degree* $\Pi_m(d_2)$ *corresponds to the maximum potential in clusters i.e.* $\Pi_m(d_2) = 0.8$.

7.10.2 Consistency procedure by computing global instances

The principle of this procedure is to check if there exists a global instance having the same possibility degree than the maximal value inside the stabilized clusters (i.e. α) without computing the global joint distribution. Indeed, we will apply the same principle than n-best stability procedure by only computing the best instances (called *best_ global_ instances*) in this joint

Table 7.22: Potentials of BCD and ABC after modification

b	c	d	π^{t+1}_{BCD}	a	b	c	π^{t}_{AC}	$\pi_{new}(BC)$	$\pi^{t+1}_{ABC} = min(\pi^{t}_{AC}, \pi_{new})$
b_1	c_1	d_1	*0*	a_1	b_1	c_1	*0.3*	*0.9*	*0.3*
b_1	c_1	d_2	*0.9*	a_1	b_1	c_2	*0.9*	*0*	*0*
b_1	c_2	d_1	***0.9***	a_1	b_2	c_1	*0.3*	*0.8*	*0.3*
b_1	c_2	d_2	*0*	a_1	b_2	c_2	*0.9*	*0.9*	*0.9*
b_2	c_1	d_1	*0*	a_2	b_1	c_1	*0.9*	*0.9*	*0.9*
b_2	c_1	d_2	***0.9***	a_2	b_1	c_2	*0.2*	*0*	*0*
b_2	c_2	d_1	*0*	a_2	b_2	c_1	*0.9*	*0.8*	*0.8*
b_2	c_2	d_2	*0.9*	a_2	b_2	c_2	*0.2*	*0.9*	*0.2*

Table 7.23: Re-stabilized potentials after incorporating $D = d_2$

a	π^S_A	a	b	π^S_{AB}	a	b	c	π^S_{ABC}	a	b	c	π^S_{ABC}	b	c	d	π^S_{BCD}	b	c	d	π^S_{BCD}
a_1	*0.4*	a_1	b_1	*0.3*	a_1	b_1	c_1	*0.3*	a_2	b_1	c_1	*0*	b_1	c_1	d_1	*0*	b_2	c_1	d_1	*0*
a_2	*0.8*	a_1	b_2	*0.4*	a_1	b_1	c_2	*0*	a_2	b_1	c_2	*0*	b_1	c_1	d_2	*0.3*	b_2	c_1	d_2	*0.8*
		a_2	b_1	*0*	a_1	b_2	c_1	*0.3*	a_2	b_2	c_1	*0.8*	b_1	c_2	d_1	*0*	b_2	c_2	d_1	*0*
		a_2	b_2	*0.8*	a_1	b_2	c_2	*0.4*	a_2	b_2	c_2	*0.2*	b_1	c_2	d_2	*0*	b_2	c_2	d_2	*0.4*

distribution using *compute_best_global_instances*. The idea is to first apply the stability procedures, then to cover all the clusters by only saving the best instances (i.e. having the maximum degree) of each of them and by combining them while eliminating the incoherent instances. In the proposed implementation, detailed in Appendix F, we use the following heuristics to determine the order in which clusters are treated to compute *best_global_instances*:

- we start by the clusters having the *less* number of best instances,
- when two or more clusters have the same number of best instances, we choose the candidate cluster with the *large* number of variables appearing in the treated cluster.

Algorithm 7.7: compute_best_global_instances

begin
  exist_global_instance $\leftarrow 1$;
  best_global_instances $\leftarrow \emptyset$;
  pos_treated_cluster $\leftarrow 1$;
  Let *nb_clusters* be the number of clusters in $\mathcal{MG}$;
  Let *big_domain* be the domain relative to all clusters;
  Let *clusters_order* be the order in which the clusters are treated (using the heuristic described above);
  treated_cluster $\leftarrow$ *clusters_order*(*pos_treated_cluster*);
  Affect the best instances in *treated_cluster* to *best_global_instances*;
  Let *treated_var* be the variable set in *treated_cluster*;
  Let *scale* be the possibilistic scale containing the possibilistic degrees in *treated_cluster*;
  if *nb_clusters*>*2* **then**
    next $\leftarrow 1$;
    pos_treated_cluster $\leftarrow 2$;
    while *(pos_treated_cluster* $\leq$ *nb_clusters) and (next = 1)* **do**
      treated_cluster $\leftarrow$ *clusters_order*(*pos_treated_cluster*);
      Combine *best_global_instances* with best instances in *treated_cluster*;
      if *incoherence* **then**
        next $\leftarrow 0$;
        exist_global_instance $\leftarrow 0$;
      else
        Update *treated_var* using variables in *treated_cluster*;
        Update *scale* using possibilistic degrees in *treated_cluster*;
        pos_treated_cluster $\leftarrow$ *pos_treated_cluster* $+ 1$;
end

This procedure stops if all clusters are treated, or if all the best instances of the treated cluster are incoherent with those of already treated clusters.

Case of consistency. If we are able to construct at least one global instance having the degree α (i.e. *exist_global_instance* $= 1$), then the moral graph is consistent. Thus, if we have already incorporated an instance of interest a, then we can take α as the exact value of $\Pi_m(a)$ i.e. $\Pi_m(a) = max\pi^C_{C_i} = \alpha$.

Case of inconsistency. If *exist_global_instance* $= 0$, then the moral graph is not yet consistent and we should decrease the maximal potential in the clusters from α to the next degree in the set of treated clusters and re-stabilize the moral graph again. The following algorithm gives the outline of this procedure:

Algorithm 7.8: Consistency procedure by computing global instances

begin
  Modification ← true;
  while Modification **do**
    Stabilize $\mathcal{MG}$ at using the simple stability procedure;
    Let α be the maximum (best) degree in the clusters;
    Let *clusters* be the set of all clusters in $\mathcal{MG}$;
    Let *best_ global_ instances* be the set of best global instances with degree α computed from *clusters* using *compute_ best_ global_ instances* ;
    Modification ← false;
    if exist_global_instance= 0 **then**
      Decrease the maximal degree in all cluster from α to the next degree in the set of treated clusters ;
      Modification←true;
end

Example 7.12 *Let us reconsider the network treated in Examples 7.7 and 7.9 . The stabilized potentials are the same than the two-parents stabilized potentials given in Tables 7.16, 7.17, 7.18, 7.19, 7.20 and 7.21. We can check that these potentials are not consistent since we are not able to construct a global instance with the degree* $\alpha_1 = 0.8215$. *Thus, the consistency procedure based on computing global instances, will decrease this degree to the following one in the scale of possibility degrees relative to all clusters (i.e. [0, 0.1911, 0.2092, 0.2104, 0.2240, 0.4538, 0.4750, 0.5064, 0.5067, 0.5088, 0.5463, 0.5797, 0.6381, 0.6707, 0.7605, 0.7646,* ***0.8161*** *0.8215]). In other terms, the value 0.8215 in the stabilized clusters will be replaced by 0.8161 and we apply the simple stability procedure again. The re-stabilized potentials are equal to the two-parents stabilized ones while replacing the value 0.8215 by 0.8161. We can check that these potentials are consistent since we are able to construct a global instance with the degree* $\alpha_1 = 0.8161$ *i.e.* $a_1 \wedge b_1 \wedge c_2 \wedge d_1 \wedge e_1 \wedge f_2 \wedge g_2$. *This means that* $\Pi_m(d_1 \wedge b_1 \wedge g_2) = 0.8161$.

7.11 Handling evidence

The proposed algorithm can be easily extended in order to take into account a new evidence e which corresponds to the value of instanciated variables. Indeed, if we are only interested with the most plausible instances of each variable in the context of the evidence e, then we should incorporate it in the moral graph and apply the propagation algorithm described above. Then, it is sufficient to take maximal instances in each cluster as the most plausible ones.

However, if we are interested with the computation of $\Pi_m(a \mid e)$, where a is the instance

of interest, then we should call the above propagation algorithm several times in order to compute successively $\Pi_m(e)$ and $\Pi_m(a \wedge e)$. Then, using the min-based conditioning, we get:

$$\Pi_m(a \mid e) = \begin{cases} \Pi_m(a \wedge e) & \text{if } \Pi_m(a \wedge e) < \Pi_m(e) \\ 1 & \text{otherwise} \end{cases}$$

Example 7.13 *Let us consider the network treated in Example 7.1. Suppose that we receive the certain information $D = d_2$. If we are only interested with the most plausible instances of each variable in the context of this evidence, then it is sufficient to compute $\Pi_m(d_2)$ as already done in Example 7.11. Indeed, from Table 7.23, we can deduce that the most plausible instance in A (resp. B, C) in the context of $D = d_2$ is a_2 (resp. b_2, c_1).*
Suppose now that we are interested with the value of $\Pi_m(a_1 \mid d_2)$. In other terms, we want to compute the impact of the evidence $D = d_2$ on the particular instance a_1 of the variable A. Then, we should first compute $\Pi_m(d_2)$ then $\Pi_m(a_1 \wedge d_2)$. Thus, we will integrate $A = a_1$ in the consistent moral graph (obtained in Example 7.11) and apply the propagation procedure again. The resulted potentials are given in Table 7.24. These potentials does not satisfy the consistency in the sense of Definition 7.5. However, the parents of all clusters having more than one parent are linked, thus no modification will be made on the moral graph and the consistency procedure stops which means that the consistency is reached. Thus, $\Pi_m(a_1 \wedge d_2) = 0.4$, which implies that $\Pi_m(a_1 \mid d_2) = 0.4$ since $\Pi_m(a_1 \wedge d_2) < \Pi_m(d_2) = 0.8$.

Table 7.24: Stabilized potentials after incorporating $D = d_2$ and $A = a_1$

a	π^S_A	a	b	π^S_{AB}	a	b	c	π^S_{ABC}	b	c	d	π^S_{BCD}
a_1	0.4	a_1	b_1	0.3	a_1	b_1	c_1	0.3	b_1	c_1	d_1	0
a_2	0	a_1	b_2	0.4	a_1	b_1	c_2	0	b_1	c_1	d_2	0.3
		a_2	b_1	0	a_1	b_2	c_1	0.3	b_1	c_2	d_1	0
		a_2	b_2	0	a_1	b_2	c_2	0.4	b_1	c_2	d_2	0
					a_2	b_1	c_1	0	b_2	c_1	d_1	0
					a_2	b_1	c_2	0	b_2	c_1	d_2	0.3
					a_2	b_2	c_1	0	b_2	c_2	d_1	0
					a_2	b_2	c_2	0	b_2	c_2	d_2	0.4

7.12 Conclusion

In this Chapter we have proposed an anytime propagation algorithm for min-based possibilistic networks. Indeed, the longer it runs, the closer to the exact marginals we get.

The proposed algorithm avoids the transformation of the initial network into a junction tree which is known to be a hard problem [30].

The principle of our propagation algorithm is to first transform the initial possibilistic graph into an equivalent undirected graph, called *moral graph*. Then, several stability procedures are proposed in order to get close to exact marginals. We study in particular three procedures,

- *Simple stability procedure*, ensuring that any cluster agrees with each of its adjacent clusters (i.e. neighbors) on the distributions defined on common variables. This procedure is polynomial.

- *Multiple nodes Stability Procedure*, ensuring that any cluster agrees on the distributions defined on common variables computed from 2, 3,.., n adjacent clusters. This procedure is also polynomial (provides that the joint distribution on considered adjacent clusters has a reasonable size).

- *Best Multiple nodes Stability Procedure*, ensuring that only best instances in the distribution of each cluster agree with the best instances in the distribution computed from several of its adjacent clusters. This procedure is polynomial in running time, however its space complexity depends on the number of best instances in the joint distribution relative to considered adjacent clusters (which can be parents, children, parents-children, neighbors).

The choice of appropriate stability procedure depends on the allowed running time and on the system capacity. Thus, we have proposed a global propagation procedure describing the best strategy in selecting stability procedures under these two constraints in order to provide best results anytime we stop the propagation process.

Finally, we have proposed two *consistency procedures* which ensures exact marginals. The first one is based on adding some links in the moral graph while the second procedure is based on constructing best global instances. These procedures are applied when stability procedure have been achieved. Note that the consistency procedure by computing global instances benefits more from stability procedures.

The consistency procedure by adding links is limited since it can enlarge some clusters with additional variables. However, the maximum number of added variables, due to an inconsistent cluster, does not exceed the maximal cardinality of parents of the variable associated with the inconsistent cluster. Moreover, this procedure can be directly applied for revising a min-based possibilistic graph by integrating a new piece of knowledge (and not simply an evidence or

observation). Namely, it can be used to construct a new DAG taking into account this new knowledge.

We should note that the results which can be provided by the anytime propagation algorithm and the classical ones are somewhat different. Indeed, classical propagation algorithms (see Chapter 6) allow to determine how any evidence on some variables affects the remaining variables, thus running them when there is no evidence seems to be meaningless. Our propagation algorithm can be used in such case and it will provide the most plausible instances relative to all variables (given initial distributions) and their possibility degree.

Appendix F contains a detailed analysis of procedures used in the implementation of our propagation algorithm.

Next Chapter shows that our propagation algorithm provides better results than the direct adaptation of junction tree algorithm (proposed in Chapter 6).

Chapter 8

Implementation and Experimentations

8.1 Introduction

In this Chapter we present the experimentation results relative to the possibilistic propagation algorithms. Indeed, using the algorithms proposed in Chapters 6 and 7, we propose a Possibilistic Networks Toolbox (**PNT**) implemented with Matlab 6.0 in order to handle possibilistic propagation. PNT proposes several propagation algorithms allowing both the product-based and the min-based propagation in possibilistic networks. The implementation of possibilistic adaptation of Pearl's algorithm and junction tree algorithm is based on the Bayes Net Toolbox (BNT) which is an open-source Matlab package for directed graphical models [99].

This Chapter is composed of two parts. Section 8.2 provides experimental results regarding the quality of our new anytime propagation algorithm with respect to exact marginals. Moreover, it studies the running time relative to different procedures of this algorithm. Section 8.3 explains the use of Possibilistic Networks Toolbox (PNT) and its main options.

8.2 Experimental results

In this Section, we provide experimental results regarding the quality of the stability procedures with respect to exact marginals. We also study the running time relative to these procedures. The experimentation is performed on random possibilistic networks generated as follows:

8.2.1 Experimental data

This Section explains which kind of graphical structures we have used during the experimental process.

Figure 8.1: Example of a DAG with 4 levels

Figure 8.2: The QMR belief network

Graphical component

Two DAG's structures have been used:

STRUCTURE 1: In this structure the DAGs are generated randomly, by just varying three parameters: the number of nodes, their cardinalities (size of their domains) and the maximum number of parents.

STRUCTURE 2: In this structure, we choose special cases of DAGs where nodes are partitioned into levels such that nodes of level i only receive arcs either from nodes of the same level, or from level $i-1$. For instance the DAG of Figure 8.1 has 4 levels: the first contains 5 nodes, the second 7 nodes, the third 3 nodes and the fourth 5 nodes.
Note that, if we consider only two levels by omitting the intra-levels links, this structure corresponds to well known networks as the QMR (Quick Medical Reference) network [77, 83] which consists of a combination of statistical and expert knowledge for approximately 600 significant diseases and approximately 4000 findings. The diseases and the findings are arranged in a bi-partite graph as shown by Figure 8.2, and the diagnosis problem is to infer a probability distribution for the diseases given a subset of findings.
An other variant of this structure can also be used in assistance for the computer network security system in the areas of intrusion detection and in particular in *partial pattern matching* and *anomaly detection* [123]. Indeed partial pattern matching can be accomplished by representing the input or symptom events (E1: event1, E2: event2, E3: event3, E4:event4) and the output or initiating events attacks (A1: attack1, A2: attack2, A3: attack3, A4: attack4) as nodes in a network as shown by Figure 8.3. Moreover, the network of Figure 8.4 is a simple example of the user modeling approach in anomaly detection task. The nodes are as follows: U: user type, A: application importance, D: high damage potential, SU: gain root privilege, M: multiple login errors, L: login error, J: joint distribution.

Figure 8.3: Belief network for pattern matching allowing for multiple attacks

Figure 8.4: Belief network for anomaly detection

Numerical component

Once the DAG structure is fixed, we generate random conditional distributions of each node in the context of its parents, respecting the normalization constraints. Then, we also generate random variable of interest.

8.2.2 Stability vs exact marginals

In the first experimentation we propose to test the quality of the stability with respect to the exact marginals $h(\pi_a)$ (i.e. $\Pi_m(a)$). Regarding the STRUCTURE 1, we have noted that the simple stability and the two-nodes stability provides, respectively, 99% and 99, 999% of exact results. That is why, we use STRUCTURE 2 considering 19 levels from 2 to 20. At each level we generate 300 networks with a number of nodes varying between 40 and 60 nodes, since we are limited, in some cases, by the junction tree algorithm[1] which is unable to treat complex networks with a great number of nodes. Table 8.1 represents different parameters for this experimentation.

Table 8.1: Parameters of the experimentation of stability vs exact marginals

levels	nodes	links	levels	nodes	links
2	45	68	12	40	83
3	40	80	13	49	106
4	45	88	14	50	107
5	40	90	15	49	103
6	40	81	16	48	99
7	40	81	17	51	106
8	40	85	18	54	112
9	40	85	19	57	119
10	40	84	20	60	125
11	40	85			

[1]The junction tree algorithm (c.f. Chapter 6) is used for providing exact values of $h(\pi_a)$.

Figure 8.5: Stability vs exact marginals

Figure 8.6: Two-nodes stability vs exact marginals

Figure 8.5, shows the results of this experimentation. At each level (from 2 to 20), the first (resp. second, third, forth) bar from the left represents the percentage of the networks where the simple stability (resp. two-nodes, three-nodes, multiple nodes, best multiple nodes) stability leads to consistency (i.e. generates the exact marginals). The values relative to two-nodes (resp. three-nodes, multiple nodes, best multiple nodes) correspond to the average of the percentages obtained with two-parents (resp. three-parents, n-parents, n-best-parents), two-children (resp. three-children, n-child, n-best-child), two-parents-children (resp. three-parents-children, n-parents-children, n-best-parents-children) and two-neighbors (resp. three-neighbors, n-neighbors, n-best-neighbors) which are represented by Figure 8.6 (resp. 8.7, 8.8, 8.9). Summary of all these percentages is given in Table 8.2.

Figure 8.7: Three-nodes stability vs exact marginals

Figure 8.8: n-nodes stability vs exact marginals

Figure 8.9: n-best-nodes stability vs exact marginals

It is clear that the higher the number of adjacent clusters considered in different stability procedures, the better the quality of results. Indeed, in multiple nodes stability procedures *n-nodes stability* is better than *three-nodes stability* which is better than *two-nodes stability* since they provide, respectively, 99.87%, 99.82%, 99.59% of exact marginals.

Moreover, Figure 8.5 shows that stability degrees, even with simple stability, are a good estimation of exact marginals (96, 42%). In addition, we remark that the quality of estimation depends on the number of levels in the DAG since with a small number of levels (2, 3 and 4), the simple stability procedure is sufficient to reach exact marginals (see Figure 8.5).

Figure 8.10 represents the running time between different stability procedures using STRUCTURE 1 with DAGs of 30 nodes and 50 links in average. It is clear that the simple stability procedure is the faster one, while the *n-nodes stability* is the slowest one. This result is unsurprising since all the stability procedures use at least one time the simple stability procedure. Moreover, *n-nodes stability* uses a greater number of clusters than *three-nodes* and *two-nodes stability*.

Figure 8.11 (resp. 8.12, 8.13, 8.14) represents details of values obtained with *two-nodes* (resp. *three-nodes*, *n-nodes*, *n-best-nodes*) stability and shows that the running time grows exponentially when using neighbors (i.e. *two-neighbors*, *three-neighbors*, *n-neighbors*, *n-best-neighbors*) which is an expected result since neighbors includes parents and children.

More generally, multiple nodes stability procedures can block if they use a great number of adjacent clusters (i.e. parents, children, parents-children or neighbors) since they are based on the computation of their joint distribution. More precisely, if there exists a cluster such that its adjacent clusters, used in the stabilization procedure, form a joint distribution with more than 7000000 instances, then the program is unable to provide an answer and blocks.

Table 8.2: Summary of the experimentation of stability vs exact marginals

Stability procedure	Percentage of exact marginals
Simple stability	**96.42%**
Two-parents	99.53%
Two-children	99.53%
Two-parents-children	99.58%
Two-neighbors	99.74%
Two-nodes stability	**99.59%**
Three-parents	99.82%
Three-children	99.82%
Three-parents-children	99.82%
Three-neighbors	99.84%
Three-nodes stability	**99.82%**
n-parents	99.84%
n-children	99.84%
n-parents-children	99.84%
n-neighbors	100%
n-nodes stability	**99.87%**
n-best-parents	99.84%
n-best-children	99.84%
n-best-parents-children	99.84%
n-best-neighbors	99.84%
n-best-nodes stability	**99.84%**

Figure 8.10: Running time between different stability procedures

From Table 8.2 we can see that *n-parents* (resp. *n-best-parents*) and *n-children* (resp. *n-best-children*) stability procedures provide the same approximation of exact marginals. However, from Figure 8.10 it is clear that *n-parents* (resp. *n-best-parents*) stability is more reasonable with respect to the running time that is why it is more efficient than *n-children* (resp. *n-best-children*) stability.

Figure 8.11: Running time between different *two-nodes* stability procedures

Figure 8.12: Running time between different *three-nodes* stability procedures

Figure 8.13: Running time between different *multiple nodes* stability procedures

8.2.3 Correlation between exact marginals and stability degrees

We are now interesting with the correlation between exact marginals and the ones generated by the stabilization procedure. This experimentation is performed on 100 random networks with 20 levels and 60 nodes (using STRUCTURE 2). Then, we compare the possibility degree of the instance of interest generated by the junction tree algorithm (exact marginals) with those generated by the simple stability procedure.

Figure 8.15 shows results of this experimentation. Again we confirm that the simple stability procedure is a good estimation of exact marginals. Indeed, it is clear that in the cases where the equality, between exact marginals and those obtained from the stability procedure, does not hold, the gap is not important. Indeed, in this experimentation the possibilistic scale contains 87 values and the gap between exact marginals and stability degrees, when they are different is equal to 5.67 values in average.

8.2.4 Comparing junction tree algorithm with the anytime algorithm

We also have compared experimentally the junction tree algorithm with simple stability procedure. In this experimentation, using STRUCTURE 1, we vary the ratio Links/Nodes in order to test the limitation of the junction tree algorithm.

This experimentation shows that with networks containing 40 (resp. 50, 60) nodes, the junction tree algorithm is blocked from the ratio 3.55 (resp. 2.72, 1.78). This is due to the size of clusters it generates in the phase of building the junction tree. Indeed, if the number of instances in any cluster contains 7000000 instances, then the algorithms is unable to initiate it.

Figure 8.14: Running time between different *Best multiple nodes* stability procedures

Figure 8.15: Correlation plots between exact marginals and somple stability procedure degrees

However, in such examples the simple stability procedure provides a result in a few seconds since it just passes messages between clusters having a limited size corresponding to each node with its parent set. Nevertheless, we should not forget that the simple stability procedure only provides good estimation of marginals but not the exact ones. Note that when the junction algorithm is not blocked, it is faster that the simple stability procedure but the difference does not exceed few seconds.

Using the exact procedure based on *computing global instances*, we can get exact marginals but this procedure can takes a long running time. For instance with networks having 60 nodes and a ratio of 1.78 this procedure provides a result in 4133,49 seconds. In other terms, computing global instances can only be limited by the running time but never blocks due to the system capacity contrary to the junction tree algorithm.

8.3 Possibilistic Networks Toolbox (PNT)

In this Section we propose a Possibilistic Networks Toolbox (**PNT**) implemented with Matlab 6.0 in order to handle possibilistic propagation. Appendix F contains a detailed analysis of procedures used to implement this software. PNT offers the following propagation algorithms:

- for min-based possibilistic networks:
  - propagation in polytrees,
  - propagation in junction trees,
  - anytime propagation.
- for product-based possibilistic networks:
  - propagation in polytrees,
  - propagation in junction trees.

The implementation of possibilistic adaptation of probabilistic propagation in polytrees and in junction trees is based on the Bayes Net Toolbox (BNT) which is an open-source Matlab package for directed graphical models [99].

The main menu of PNT is represented by Figure 8.16. We now explain its different options.

Figure 8.16: Main menu

8.3.1 Definition of network structure

This option (see Figure 8.17) allows the specification of the DAG structure by adding or deleting nodes or links. The addition of any node is done by choosing its position, its name and its parent set within existing nodes.
For instance Figures 8.18, 8.19, 8.20 and 8.21 represent different steps relative to the addition of the the node D to the DAG relative to the possibilistic network used in Example 7.1.

Figure 8.17: Definition of the network structure

Figure 8.18: Choice of the position of the node D

8.3.2 Network Quantification

This option allows the quantification a constructed network by providing the cardinality and the initial distributions relative to each node. Moreover it allows the definition of the observed nodes and the variable of interest.

Definition of cardinalities and initial distributions

Given a DAG structure, this option (see Figure 8.22) allows its quantification. The first step is to define the cardinality of each variable in order to construct the network structure. The second step is to provide the initial local conditional possibility distribution of each variable in the context of its parents.

Figures 8.23, 8.24 and 8.25 are relative to the quantification of the possibilistic network of Example 7.1.

Figure 8.19: Definition of the name of the new node (D)

Figure 8.20: Selection of the parents of D within A, B and C ($U_D = \{B, C\}$)

Definition of evidence and instance of interest

The definition of total evidence can be done via *Define evidence* option (see Figure 8.26) by selecting the observed variables and their certain instances. In the same manner, *Define Instance of Interest* option (see Figure 8.27) allows the definition the instance of interest by selecting variables of interest. Note that the definition of evidence is necessary except with the anytime propagation algorithm. Indeed, propagation algorithms in junction trees and in polytrees are meaningless if the evidence is not specified.

Figures 8.28 and 8.29 are relative to the evidence $D = d_2$ and the instance of interest $A = a_1$.

8.3.3 Min-based propagation

Given a product-based possibilistic network and the set of observed nodes, we can call any of the following propagation algorithms.

Propagation in polytrees

The min-based propagation in polytrees (see Figure 8.31) is based in the algorithm presented in Section 6.3.1. If we call this algorithm with a multiply connected DAG, then an error message is displayed. Otherwise, the propagation result is provided.

Figure 8.21: DAG of Example 7.1

Figure 8.22: Definition of cardinalities and initial distributions

Figure 8.23: Example of definition of cardinalities

Figure 8.24: Example of initial conditional distributions

Figure 8.25: Example of initial conditional distributions

Figure 8.26: Definition of evidence

Figure 8.27: Definition of instance of interest

Figure 8.28: Selection of the observed node D and its instance d_2

Figure 8.29: Selection of the variable of interest A and its instance a_1

Figure 8.30: Coloration of the observed variable D (blue) and the variable of interest A (red)

Figure 8.31: Min-based propagation in polytrees

Figure 8.32: Min-based propagation in junction trees

Propagation in junction trees

The min-based propagation algorithm in junction trees (see Figure 8.32) is based in the algorithm presented in Section 6.3.2.

Anytime algorithm

Once the network is defined, the anytime propagation algorithm (see Figure 8.33) provides results of propagation which depend on the definition of evidence and instance of interest. Indeed,

- if the evidence and the instance of interest are specified, then the propagation algorithm displays the conditional possibility degree of the instance of interest in the context of evidence,
- if only the evidence is specified, then it displays the most plausible instances relative to the remaining variables i.e. those which are not observed,
- if only the instance of interest is specified, then it displays the possibility degree of this instance (with no context),
- if neither the evidence nor the instance of interest are specified, then it displays the most plausible instances relative to all variables (given initial distributions).

We can also choose the procedures to apply during the propagation process. In this case, we should specify if we want to apply the consistency procedure or not (see Figure 8.35). If it is the case then we should select one of the two proposed consistency procedures (i.e. add links or compute best global instances) (see Figure 8.34). Otherwise, we should specify the number and the type of nodes to consider in the selected stability procedure (see Figures 8.34 and 8.36)).

Figure 8.37 shows the propagation result relative to the evidence $D = d_2$ and the instance of interest $A = a_1$ (i.e. $\Pi_m(a_1 \mid d_2)$) using a simple stability procedure. Moreover, Figure 8.38 shows the propagation result if we run the propagation algorithm with neither evidence nor instance of interest (which corresponds to the result obtained in Example 7.6).

Figure 8.33: Anytime Propagation

Figure 8.34: Definition of consistency options

Figure 8.35: Definition of stability options

Figure 8.36: Definition of stability options

Figure 8.37: Value of $\Pi_m(a_1 \mid d_2) = 0.4$

Figure 8.38: Most plausible instances

Figure 8.39: Product-based propagation in polytrees

Figure 8.40: Product-based propagation in junction trees

8.3.4 Product-based propagation

Given a product-based possibilistic network and the set of observed nodes, we can call any of the following propagation algorithms.

Propagation in polytrees

The min-based propagation in polytrees (see Figure 8.39) is based in the algorithm presented in Section 6.2.1. If we call this algorithm with a multiply connected DAG, then an error message is displayed. Otherwise, the propagation result is provided.

Propagation in junction trees

The min-based propagation algorithm in junction trees (see Figure 8.40) is based in the algorithm presented in Section 6.2.2.

8.4 Conclusion

The experimentation results, provided in this Chapter, show that our new algorithm (described in Chapter 7) gives better results than a direct adaptation of probabilistic propagation algorithm.

Indeed, the simple stability procedure can be applied efficiently to any DAG structure. Moreover, it provides a high number of exact marginals (i.e. 96.42%). Other refined stability procedures improve the rate of correct exact marginals (for instance *n-nodes stability* provides 99.87% of exact marginals), without a huge increasing of the running time (with a DAG having 60 nodes, the additional running time is between 10 and 60 seconds).

These results are interesting since they show that with networks having complex structures with a great number of nodes, we can use simple stability procedure which is a *polynomial procedure*. Indeed, in such cases junction tree algorithm can generate huge clusters where local computations are impossible, and blocks.

The second part of this Chapter has illustrated our implementation of different propagation algorithms for min-based and product-based possibilistic networks and have shown the use of our Possibilistic Networks Toolbox (**PNT**).

Conclusion

This thesis has contributed to the development of graphical models for reasoning under uncertainty in possibility theory framework. Uncertainty is either encoded *numerically* using the unit interval [0,1] or *qualitatively* using a total pre-order between events.

We have addressed the question of defining independence relations in possibility theory. We have first introduced a qualitative framework where uncertainty is represented by total pre-orders on possible states of the universe of discourse. This framework recovers basic definitions of classical possibility theory. For instance, the qualitative conditioning extends standard possibilistic conditioning.

Then, we have noticed that there are two kinds of independence: *causal* and *decompositional*. Causal independence relations can be simply defined using notions of accepted, ignored and rejected beliefs. Decompositional independence relations are defined using other operators different from the two traditional ones: *minimum* and *product* operators. These two kinds of independence are equivalent in probability theory, while in possibilistic setting we only have decompositional independence which implies causal ones.

Our main contribution regarding independence relations is the proposition of new definitions obtained from the analysis of the plausibility ordering induced from a possibility distribution. Obviously, these new definitions have been compared with existing ones and their graphoid properties have been studied.

Our results on independence relations can be used in multiple criteria analysis and in relational data decomposition. They can also be used for defining other forms of qualitative networks. For instance, Brafmann and col. [27] have proposed a new qualitative network where inside each node a plausibility relation is used instead of possibility degrees. They use *Ceteris Paribus* independence (i.e. CP-independence) which is, as we have shown, equivalent to the qualitative independence relation based on preserving orderings (i.e. POS-independence).

Therefore, our study of graphoid properties can be useful for showing the coherence of propagation algorithms based on Ceteris Paribus independence.

Another main contribution of this thesis is the development of graphical models for possibility theory framework. We have first developed a direct adaptation of the centralized version of Pearl's algorithm [87, 103, 105] and of the probabilistic propagation in junction trees [84] for min-based and product-based possibilistic networks. This allows us to conclude that when we use the product form of conditioning, we get possibilistic networks close to the probabilistic ones, sharing the same features and having the same theoretical and practical results. It is also important to note that other existing graphical models like *Valuation Based Systems* [111, 112] or networks based on Spohn's ordinal conditional functions [35, 72] only allow to recover product-based networks and not the min-based ones. In fact, min-based networks have different behavior since they do not satisfy the so-called *coherence* property. Indeed, it may happen that the joint distribution associated with the possibilistic graph do not recover the initial data provided by experts. Nevertheless, we have shown that unrecovered data have no effect either on independence relations or on the propagation process.

The particular properties of the minimum operator, such as the idempotency, lead us to explore a new propagation approach for min-based networks which avoids the transformation of the initial network into a junction tree, known to be a hard problem.

The proposed algorithm is an *anytime* algorithm. It is composed of several steps, which progressively converge to the exact marginals. The experimentation results show that our new algorithm gives better results than a direct adaptation of the probabilistic propagation algorithm. Indeed, the simple stability procedure can be applied efficiently to any DAG structure, including those where the junction tree algorithm blocks. Moreover, it provides a high number of exact marginals (i.e. 96.42%). Other refined stability procedures improve the rate of correct exact marginals (for instance *n-nodes stability* provides 99.87% of exact marginals), without a huge increasing of running time (with a DAG having 60 nodes, the additional running time is between 10 and 60 seconds).

We have also proposed a Possibilistic Networks Toolbox (**PNT**) implemented with Matlab 6.0, in order to handle possibilistic propagation in both product-based and min-based possibilistic networks.

An interesting future work is to compare the graphical-based representation of uncertain information with the logic-based one. Recently, a study of the links existing between possibilistic logic and possibilistic graphical models has been proposed by Benferhat and col. [17, 18]. This work provides theoretical results to encode possibilistic logic bases into possibilistic graphs. It also studies the translation of possibilistic graphs into possibilistic logic. Such results can be interesting to take the advantage of both frameworks. Nevertheless, the complexity of proposed methods is not yet performed.

From application point of view, qualitative possibilistic networks can be used in assistance for the computer network security system in the areas of *intrusion detection* [123]. Indeed, in such applications qualitative results, for instance on the *attack nature* are expected rather than exact numerical values. Besides, the responding time is crucial due to the danger which can follow an attack of the computer system.

Another line of research will be to handle decision variables by studying *possibilistic influence diagrams*. Indeed, most of the proposed methods for propagation in probabilistic influence diagrams are based on propagation on probabilistic Bayesian networks [109, 132]. The same idea can be used on possibility theory by using propagation results obtained in this thesis. Such a development needs the *qualitative decision* concepts recently presented in possibility theory [51, 59, 131].

Appendix A

Proofs of Chapter 2

Proof of Proposition 2.1 We want to prove that
(i) $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) \neq min(\mathbf{Acc}_{\geq_\pi}(\phi), \mathbf{Acc}_{\geq_\pi}(\psi))$
$\Leftrightarrow \mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = -1, \mathbf{Acc}_{\geq_\pi}(\phi) = 0,$ and $\mathbf{Acc}_{\geq_\pi}(\psi) = 0.$
Let us consider the possible values of $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi)$:

- $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = 1 \Rightarrow \mathbf{Acc}_{\geq_\pi}(\phi) = 1$ and $\mathbf{Acc}_{\geq_\pi}(\psi) = 1$ (using property 2)
  Hence, $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = min(\mathbf{Acc}_{\geq_\pi}(\phi), \mathbf{Acc}_{\geq_\pi}(\psi))$.

- $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = 0 \Rightarrow \exists \phi', \psi'$ s.t. $\phi \wedge \psi =_\pi \phi' \wedge \psi'$ (at least $\phi' \neq_\Pi \phi$ or $\psi' \neq_\Pi \psi$)
  If we assume that $\phi' \neq_\Pi \phi$, then $\mathbf{Acc}_{\geq_\pi}(\phi) = 0$ and $\mathbf{Acc}_{\geq_\pi}(\psi) \geq 0$
  $\Rightarrow \mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = min(\mathbf{Acc}_{\geq_\pi}(\phi), \mathbf{Acc}_{\geq_\pi}(\psi))$.

- $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = -1$. Using (i), we deduce that (i1) $\mathbf{Acc}_{\geq_\pi}(\phi) \geq 0$ and (i2) $\mathbf{Acc}_{\geq_\pi}(\psi) \geq 0$.
  If we assume that $\mathbf{Acc}_{\geq_\pi}(\phi) = 1$, then $\phi >_\pi \neg\phi$. Thus, we can distinguish two cases:
  - $\phi \wedge \neg\psi$ is more plausible than $\neg\phi \wedge \psi$ and $\neg\phi \wedge \neg\psi$
    $\Rightarrow \mathbf{Acc}_{\geq_\pi}(\psi) = -1$ which contradicts (i2).
  - $\phi \wedge \psi$ is more plausible than $\neg\phi \wedge \psi$ and $\neg\phi \wedge \neg\psi$. In this case $\phi \wedge \neg\psi$ is more plausible than $\phi \wedge \psi$ (since $\mathbf{Acc}_{\geq_\pi}(\phi \wedge \psi) = -1$) $\Rightarrow \mathbf{Acc}_{\geq_\pi}(\psi) = -1$ which contradicts (i2).

Proof of Proposition 2.2 Let ϕ and ψ be subclasses of Ω, we need to prove that:
$\mathbf{Acc}(\phi \wedge \psi) = min(\mathbf{Acc}(\phi \mid \psi), \mathbf{Acc}(\psi))$. The possible values that $\mathbf{Acc}(\phi \wedge \psi)$ can take on are:

- $\mathbf{Acc}(\phi \wedge \psi) = 1 \Rightarrow \phi \wedge \psi >_\Pi \neg(\phi \wedge \psi) = max(\neg\phi \wedge \psi, \phi \wedge \neg\psi, \neg\phi \wedge \neg\psi)$ and hence $\phi \wedge \psi >_\Pi \neg\phi \wedge \psi$ which implies $\mathbf{Acc}(\phi \mid \psi) = 1$.

- $\mathbf{Acc}(\phi \wedge \psi) = -1 \Rightarrow \phi \wedge \psi <_\Pi \neg(\phi \wedge \psi)$.
  $\Rightarrow max(\Omega) \subseteq (\neg\phi \wedge \psi) \vee (\phi \wedge \neg\psi) \vee (\neg\phi \wedge \neg\psi)$,
  - if $\neg\phi \wedge \psi \subseteq max(\Omega)$ then $\mathbf{Acc}(\phi \mid \psi) = -1$,
  - else $max(\Omega) \subseteq \neg\psi$, $\psi <_\Pi \neg\psi$ which implies that $\mathbf{Acc}(\psi) = -1$.

- $\mathbf{Acc}(\phi \wedge \psi) = 0 \Rightarrow \phi \wedge \psi =_\Pi \neg(\phi \wedge \psi)$
  $\Rightarrow \phi \wedge \psi \subseteq max(\Omega)$ and $(\neg\phi \wedge \psi) \vee (\phi \wedge \neg\psi) \vee (\neg\phi \wedge \neg\psi) \cap max(\Omega) \neq \emptyset$,
  - if $\neg\phi \wedge \psi \subseteq max(\Omega) \Rightarrow \mathbf{Acc}(\phi \mid \psi) = 0$ (since $\phi \wedge \psi =_\Pi \neg\phi \wedge \psi$)
  and $\mathbf{Acc}(\psi) \geq 0$ (since $\phi \wedge \psi \subseteq \psi$),
  - else $max(\Omega) \subseteq \neg\phi \Rightarrow \mathbf{Acc}(\phi \mid \psi) = 1$ (since $\phi \wedge \psi >_\Pi \neg\phi \wedge \psi$)
  and $\mathbf{Acc}(\psi) = 0$ (since $\psi =_\Pi \max_{\omega \in [\psi]} \omega =_\Pi \neg\psi =_\Pi \max_{\omega \in [\neg\psi]} \omega$).

Appendix B

Proofs of Chapter 3

For the sake of simplicity, the context appearing behind the conditioning bar is omitted in the some of the following proofs.

B.1 Proofs for independence relations

Proof of Proposition 3.1 Let $\geq_\pi$ be a plausibility relation defined on $\Omega = D_V$ and consider three mutually disjoint subsets of variables X, Y and Z forming a partition of V. We want to prove that the relation $I_{BP}(X, Z, Y)$ is true, iff, $\forall x, y, z, \mathbf{Acc}(x \mid y \wedge z) = \mathbf{Acc}(x \mid z)$. The independence relation $I_{BP}(X, Z, Y)$ implies this Proposition trivially by replacing ϕ_X by x, ψ_Y by y and φ_Z by z in (3.1). Thus it is enough to prove that if $\forall x, y, z, \mathbf{Acc}(x \mid y \wedge z) = \mathbf{Acc}(x \mid z)$ then $I_{BP}(X, Z, Y)$. Since **Acc** on instances of X characterizes the acceptance function on subsets of D_X, it follows that $\forall y, z$ $\mathbf{Acc}(\phi_X \mid y \wedge z) = \mathbf{Acc}(\phi_X \mid z)$. Now it is obvious that plausibility measure relations $\psi_1 >_\Pi \varphi_1$ and $\psi_2 >_\Pi \varphi_2$ imply $\psi_1 \vee \psi_2 >_\Pi \varphi_1 \vee \varphi_2$ and the same for $=_\Pi$. Hence, from $\phi_X \wedge y \wedge z >_\Pi \neg\phi_X \wedge y \wedge z$ $\forall y \in \psi_Y, \forall z \in \varphi_Z$ imply $\phi_X \wedge \psi_Y \wedge \varphi_Z >_\Pi \neg\phi_X \wedge \psi_Y \wedge \varphi_Z$, and the same with $=_\Pi$. Hence, $I_{BP}(X, Z, Y)$ holds.

Proof of Proposition 3.2 We want to prove that if X is PO-independent of Y, then X is also BP-independent of Y. We can distinguish three cases:
- $\mathbf{Acc}(x) = 1$, this means that $\forall x', x >_\Pi x'$. Therefore, $\forall x', x \wedge y >_\pi x' \wedge y$, $\mathbf{Acc}(x \mid y) = 1$.
- $\mathbf{Acc}(x) = -1$, this means that $\exists x'$ s.t. $x' >_\Pi x$. Then using PO-independence we deduce that $\exists x'$ s.t. $x' \wedge y >_\pi x \wedge y$ which implies $\mathbf{Acc}(x \mid y) = -1$
- $\mathbf{Acc}(x) = 0$, this means that $\exists x'$ s.t. $x =_\Pi x'$ and $\nexists x''$ s.t. $x'' >_\Pi x$
$\Rightarrow x \wedge y =_\pi x' \wedge y$ and $\nexists x''$ s.t. $x'' \wedge y =_\pi x \wedge y$
$\Rightarrow \mathbf{Acc}(x \mid y) = 0$.

Proof of Proposition 3.3 Let $\phi = D'_X$ and $\psi = D'_Y$.
- Firstly we prove that if $\forall \phi \subseteq D_X, \forall \psi \subseteq D_Y$ s.t. $\phi \neq \emptyset$ and $\psi \neq \emptyset$, $\forall x, y$,
$\mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) = min(\mathbf{Acc}(x \mid \phi), \mathbf{Acc}(y \mid \psi))$ then X is POS-independent of Y.
Assume that (i) $\mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) = min(\mathbf{Acc}(x \mid \phi), \mathbf{Acc}(y \mid \psi))$
then $\forall x, x', \forall y$, we have:
$\mathbf{Acc}(x \wedge y \mid \{x, x'\} \wedge y) = min(\mathbf{Acc}(x \mid \{x, x'\}), \mathbf{Acc}(y \mid y))$
$\Leftrightarrow \mathbf{Acc}(x \mid \{x, x'\} \wedge y) = \mathbf{Acc}(x \mid \{x, x'\})$
since $\mathbf{Acc}(y \mid y) = 1$, and $\mathbf{Acc}(x \wedge y \mid \{x, x'\} \wedge y) = \mathbf{Acc}(x \mid \{x, x'\} \wedge y)$.
If $\mathbf{Acc}(x \mid \{x, x'\}) = 1$ (resp. 0,-1), then $x >_\Pi x'$ (resp. $x =_\Pi x', x' >_\Pi x$)
$\Rightarrow \mathbf{Acc}(x \mid \{x, x'\} \wedge y) = 1$ (resp. 0,-1)
$\Rightarrow x \wedge y >_\pi x' \wedge y$ (resp. $x \wedge y =_\pi x' \wedge y, x' \wedge y >_\pi x \wedge y$).
Therefore, X is PO-independent of Y, Moreover Y is PO-independent of X since (i) is obviously symmetric. Hence, X is POS-independent of Y.

- Let us show the converse. Assume that X and Y are POS-independent.
First suppose that $\mathbf{Acc}(x \mid \phi) = -1$. Then $\exists x' \in \phi$ such that $x' >_\Pi x$
$\Rightarrow x' \wedge y >_\pi x \wedge y$ (since I_{POS} is true)
$\Rightarrow \mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) = -1$ (by definition). The same conclusion holds if $\mathbf{Acc}(y \mid \psi) = -1$.
Now, we have three remaining cases (other cases, are obtained by symmetry):

- $\mathbf{Acc}(x \mid \phi) = 1$ and $\mathbf{Acc}(y \mid \psi) = 1$.
  Assume that $\mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) \neq 1$ (i.e. 0 or -1)
  $\Rightarrow \exists x' \in \phi, y' \in \psi$ such that $x' \wedge y' \geq_\pi x \wedge y$
  Moreover, $x >_\Pi x'$ (since $\mathbf{Acc}(x \mid \phi) = 1$)
  $\Rightarrow x \wedge y >_\pi x' \wedge y$ (since I_{POS} is true)
  $\Rightarrow x' \wedge y' \geq_\pi x \wedge y >_\pi x' \wedge y$
  $\Rightarrow x' \wedge y' >_\pi x' \wedge y$
  $\Rightarrow y' \geq_\Pi y$(since I_{POS} is true). Hence contradiction with $\mathbf{Acc}(y \mid \psi) = 1$.

- $\mathbf{Acc}(x \mid \phi) = 0$ and $\mathbf{Acc}(y \mid \psi) = 1$.
  Assume that $\mathbf{Acc}(x \wedge y \mid \phi, \psi) \neq 0$ (i.e. -1 or 1), then we can consider two subcases:
  - $\mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) = -1 \Rightarrow \exists x' \in \phi, y' \in \psi$ such that $x' \wedge y' >_\pi x \wedge y$
  Moreover, $y >_\Pi y'$ (since $\mathbf{Acc}(y \mid \psi) = 1$)
  $\Rightarrow x \wedge y >_\pi x \wedge y'$ (since I_{POS} is true)
  $\Rightarrow x' \wedge y' >_\pi x \wedge y'$
  $\Rightarrow x' >_\Pi x$ (since I_{POS} is true). Hence contradiction with $\mathbf{Acc}(x \mid \phi) = 0$.
  - $\mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) = 1 \Rightarrow \forall x' \in \phi, \forall y' \in \psi, x \wedge y >_\pi x' \wedge y'$

$\Rightarrow \forall x', x \wedge y >_\pi x' \wedge y \Rightarrow x >_\Pi x'$ (since I_{POS} is true).
Hence contradiction with $\mathbf{Acc}(x \mid \phi) = 0$.

- $\mathbf{Acc}(x \mid \phi) = 0$ and $\mathbf{Acc}(y \mid \psi) = 0$.
  Assume that $\mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) \neq 0$ (i.e. -1 or 1), then we can consider two subcases:
  - $\mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) = -1 \Rightarrow \exists x' \in \phi, y' \in \psi$ such that $x' \wedge y' >_\pi x \wedge y$
  Moreover, $y =_\Pi y'$ (since $\mathbf{Acc}(y \mid \psi) = 0$)
  $\Rightarrow x \wedge y =_\pi x \wedge y'$ (since I_{POS} is true)
  $\Rightarrow x' \wedge y' >_\pi x \wedge y'$
  $\Rightarrow x' >_\Pi x$(since I_{POS} is true). Hence contradiction with $\mathbf{Acc}(x \mid \phi) = 0$.
  - $\mathbf{Acc}(x \wedge y \mid \phi \wedge \psi) = 1 \Rightarrow \forall x' \in \phi, \forall y' \in \psi, x \wedge y >_\pi x' \wedge y'$
  $\Rightarrow \forall x', x \wedge y >_\pi x' \wedge y \Rightarrow \forall x', x >_\Pi x'$ (since I_{POS} is true).
  Hence contradiction with $\mathbf{Acc}(x \mid \phi) = 0$.

Proof of Proposition 3.4 We want to prove that POS-independence implies the BPS-independence. This relation is true. Indeed, if we let $D'_X = D_X$ and $D'_Y = \{y\}$ in (3.6) we obtain:
$\mathbf{Acc}(x \wedge y \mid D_X, \{y\}) = min(\mathbf{Acc}(x \mid D_X), \mathbf{Acc}(y \mid \{y\}))$
$\Leftrightarrow \mathbf{Acc}(x \mid \{y\}) = \mathbf{Acc}(x)$ (since $\mathbf{Acc}(x \wedge y \mid D_X, \{y\}) = \mathbf{Acc}(x \mid \{y\})$,
$\mathbf{Acc}(x \mid D_X) = \mathbf{Acc}(x)$ and $\mathbf{Acc}(y \mid \{y\}) = 1$)
which leads to case (i) of (3.3). The case (ii) of (3.3) is obtained by symmetry by letting $D'_X = \{x\}$ and $D'_Y = D_Y$ in (3.6).

Proof of Proposition 3.5 We want to prove that CP-independence is equivalent to the POS-independence.

- Suppose that X and Y are CP-independent but not POS-independent i.e.
  $x_i > x_j$ but $\exists y$ s.t. $x_i \wedge y \leq x_j \wedge y$
  From CP-independence $x_i \wedge y \leq x_j \wedge y \Rightarrow \forall y', x_i \wedge y' \leq x_j \wedge y'$
  $\Rightarrow max\{x_i \wedge y'\} \leq max\{x_i \wedge y'\} \Rightarrow x_i \leq x_j$
  Hence contradiction.

- Suppose that X and Y are POS-independent but not CP-independent i.e.
  $x_i \wedge y > x_j \wedge y$ but $\exists y'$ s.t. $x_i \wedge y' \leq x_j \wedge y'$
  The relation $x_i \wedge y > x_j \wedge y$ implies $x_i > x_j$,
  $\Rightarrow x_i \wedge y' > x_j \wedge y'$ (From POS-independence)
  Hence contradiction.

Proof of Proposition 3.6 We want to prove that X and Y are PT-independent in the context Z as soon as Definition 3.4 holds for instances only. To see it, note that (3.9) does not hold only if $\mathbf{Acc}(x \wedge y \mid z) = -1$ and $\mathbf{Acc}(x \mid z) = \mathbf{Acc}(y \mid z) = 0$ due to the properties of **Acc** given in Subsection 2.4.
This case can only be observed if the set of plausible instances of X (resp. Y) contains more than one element, including x (resp. y). So, $x \in max(D_X), y \in max(D_Y), \mid D_X \mid > 1$ and $\mid D_Y \mid > 1$, and (3.9) does not hold if $x \wedge y$ is not a plausible instance in $D_X \times D_Y$. So (3.9) means that $max(D_X \times D_Y) = max(D_X) \times max(D_Y)$ in any context z. Hence $\forall \varphi_Z \subseteq D_Z, max(D_X \times D_Y) \times D_Z = max(D_X) \times max(D_Y) \times D_Z$. However, Definition (3.4) does not apply only if:
$\mathbf{Acc}(\phi_X \wedge \psi_Y \mid \varphi_Z) = -1, \mathbf{Acc}(\phi_X \mid \varphi_Z) = 0,$ and $\mathbf{Acc}(\psi_Y \mid \varphi_Z) = 0.$
It is equivalent to say that:
$\phi_X \wedge \neg\psi_Y \wedge \varphi_Z =_\pi \neg\phi_X \wedge \psi_Y \wedge \varphi_Z \geq_\pi \neg\phi_X \wedge \neg\psi_Y \wedge \varphi_Z >_\pi \phi_X \wedge \psi_Y \wedge \varphi_Z.$

However, it implies that $max(D_X)$ overlaps ϕ_X and $\neg\phi_X$, $max(D_Y)$ overlaps ψ_Y and $\neg\psi_Y$, and $\forall x \wedge y \in \phi_X \wedge \psi_Y \wedge max(D_X) \times max(D_y)$, $x \wedge y \notin max(D_X \times D_Y)$. Hence, we have proved that in context φ_Z, the equality $\mathbf{Acc}(x \wedge y \mid \varphi_Z) = min(\mathbf{Acc}(x \mid \varphi_Z), \mathbf{Acc}(y \mid \varphi_Z))$ does not hold. It implies that $\exists z \in \varphi_Z$, such that (3.9) does not hold. So (3.9) implies PT-independence.

Proof of Proposition 3.7 We want to prove that the acceptance of one instance of X or of Y is enough to conclude PT-independence between these two variable sets in the context Z. More formally, we want to prove that $\forall z \in D_Z$,
if $\exists x \in D_X$ s.t. $\mathbf{Acc}(x \mid z) = 1$, or if $\exists y \in D_Y$ s.t. $\mathbf{Acc}(y \mid z) = 1$,
then the relation $I_{PT}(X, Z, Y)$ is true.

We analyze the possible situations (the other are obtained by symmetry):
(a) $\mathbf{Acc}(x \mid z) = 1$ and $\mathbf{Acc}(y \mid z) = 0 \Rightarrow min(\mathbf{Acc}(x \mid z), \mathbf{Acc}(y \mid z)) = 0$
(b) $\mathbf{Acc}(x \mid z) = 1$ and $\mathbf{Acc}(y \mid z) = 1 \Rightarrow min(\mathbf{Acc}(x \mid z), \mathbf{Acc}(y \mid z)) = 1$
(c) $\mathbf{Acc}(x \mid z) = 1$ and $\mathbf{Acc}(y \mid z) = -1 \Rightarrow min(\mathbf{Acc}(x \mid z), \mathbf{Acc}(y \mid z)) = -1$

- In the case (a) we have $x \wedge z >_\Pi \neg x \wedge z$ and $y \wedge z =_\Pi \neg y \wedge z$
  $\Rightarrow max(x \wedge y \wedge z, x \wedge \neg y \wedge z) >_\pi max(\neg x \wedge y \wedge z, \neg x \wedge \neg y \wedge z)$ and
  $max(x \wedge y \wedge z, \neg x \wedge y \wedge z) =_\pi max(x \wedge \neg y \wedge z, \neg x \wedge \neg y \wedge z)$
  $\Rightarrow x \wedge y \wedge z =_\pi x \wedge \neg y \wedge z >_\pi max(\neg x \wedge y \wedge z, \neg x \wedge \neg y \wedge z)$
  $\Rightarrow \mathbf{Acc}(x \wedge y \mid z) = 0$

- In the case (b) we have $x \wedge z >_\Pi \neg x \wedge z$ and $y \wedge z >_\Pi \neg y \wedge z$

$\Rightarrow max(x \wedge y \wedge z, x \wedge \neg y \wedge z) >_\pi max(\neg x \wedge y \wedge z, \neg x \wedge \neg y \wedge z)$ and
$max(x \wedge y \wedge z, \neg x \wedge y \wedge z) >_\pi max(x \wedge \neg y \wedge z, \neg x \wedge \neg y \wedge z)$
$\Rightarrow x \wedge y \wedge z >_\pi max(\neg x \wedge y \wedge z, x \wedge \neg y \wedge z, \neg x \wedge \neg y \wedge z)$
$\Rightarrow \mathbf{Acc}(x \wedge y \mid z) = 1$

- In the case (c) we have $x \wedge z >_\Pi \neg x \wedge z$ and $\neg y \wedge z >_\Pi y \wedge z$
  $\Rightarrow max(x \wedge y \wedge z, x \wedge \neg y \wedge z) >_\pi max(\neg x \wedge y \wedge z, \neg x \wedge \neg y \wedge z)$ and
  $max(x \wedge \neg y \wedge z, \neg x \wedge \neg y \wedge z) >_\pi max(x \wedge y \wedge z, \neg x \wedge y \wedge z)$
  $\Rightarrow x \wedge \neg y \wedge z >_\pi max(\neg x \wedge y \wedge z, x \wedge y \wedge z, \neg x \wedge \neg y \wedge z)$
  $\Rightarrow \mathbf{Acc}(x \wedge y \mid z) = -1$

Proof of Proposition 3.8 We want to prove that,
(i) $\mathbf{Acc}(x \wedge y) \neq min(\mathbf{Acc}(x), \mathbf{Acc}(y)) \Leftrightarrow \mathbf{Acc}(x \wedge y) = -1, \mathbf{Acc}(x) = 0,$ and $\mathbf{Acc}(y) = 0.$
Let us consider the different cases of $\mathbf{Acc}(x \wedge y)$:

- $\mathbf{Acc}(x \wedge y) = 1 \Rightarrow x \wedge y >_\pi x' \wedge y', \forall x' \in D_X \setminus \{x\}, \forall y' \in D_Y \setminus \{y\}$
  $\Rightarrow x >_\Pi x'$ and $y >_\Pi y', \forall x' \in D_X \setminus \{x\}, \forall y' \in D_Y \setminus \{y\}$
  $\Rightarrow \mathbf{Acc}(x) = 1$ and $\mathbf{Acc}(y) = 1$
  Hence, $\mathbf{Acc}(x \wedge y) = min(\mathbf{Acc}(x), \mathbf{Acc}(y))$.
  Thus, this case is impossible since it contradicts (i).

- $\mathbf{Acc}(x \wedge y) = 0 \Rightarrow \exists x', y'$ s.t. $x \wedge y =_\pi x' \wedge y'$ (at least $x' \neq_\Pi x$ or $y' \neq_\Pi y$)
  If we assume that $x' \neq_\Pi x$, then $\mathbf{Acc}(x) = 0$ and $\mathbf{Acc}(y) \geq 0$
  $\Rightarrow \mathbf{Acc}(x \wedge y) = min(\mathbf{Acc}(x), \mathbf{Acc}(y))$. Thus, this case is impossible since it contradicts (i).

- $\mathbf{Acc}(x \wedge y) = -1$. Then, using (i) we deduce that (i1) $\mathbf{Acc}(x) \geq 0$ and (i2) $\mathbf{Acc}(y) \geq 0$.
  If we assume that $\mathbf{Acc}(x) = 1$, then we can distinguish two cases:

  - $x \wedge \neg y$ is more plausible than $\neg x \wedge y$ and $\neg x \wedge \neg y$
    $\Rightarrow \mathbf{Acc}(y) = -1$ which contradicts (i2).
  - $x \wedge y$ is more plausible than $\neg x \wedge y$ and $\neg x \wedge \neg y$. In this case $x \wedge \neg y$ is more plausible than $x \wedge y$ (since $\mathbf{Acc}(x \wedge y) = -1$) $\Rightarrow \mathbf{Acc}(y) = -1$ which contradicts (i2).

Proof of Proposition 3.9 We want to prove that if X and Y are BPS-independent then they are PT-independent. Suppose that X and Y are BPS but not PT-independent:
$\Rightarrow \exists x, \exists y$, s.t. $\mathbf{Acc}(x \wedge y) \neq min(\mathbf{Acc}(x), \mathbf{Acc}(y))$
$\Rightarrow min(\mathbf{Acc}(x \mid y), \mathbf{Acc}(y)) \neq min(\mathbf{Acc}(x), \mathbf{Acc}(y))$ (From Proposition 2.2)
Hence $\mathbf{Acc}(x \mid y) \neq \mathbf{Acc}(x)$.

So X and Y are not PO-independent. Hence they cannot be BPS-independent.

Proof of Proposition 3.10 We want to prove that if a plausibility relation $\geq_\pi$ is Pareto-decomposable along X and Y then one of the local plausibility relations on X or Y should be *uniform*. Suppose that none of the distributions on X and Y is uniform, that is $\exists x, x', \exists y, y'$, s.t. $x >_\Pi x'$ and $y >_\Pi y'$. Then the two states $x \wedge y'$ and $x' \wedge y$ are not comparable. Indeed:
- if $x \wedge y' \geq_\pi x' \wedge y$, this relation contradicts the Pareto-ordering since $y' \not\geq_\Pi y$,
- if $x' \wedge y \geq_\pi x \wedge y'$, this relation contradicts the Pareto-ordering since $x' \not\geq_\Pi x$.
This result contradicts the assumption that $\geq_\pi$ encodes a complete preorder.

Proof of Proposition 3.11
- PROOF THAT I_{Pareto} IMPLIES $I_{leximin}$. Suppose that X and Y are Pareto-independent but not leximin-independent, then we can distinguish two cases:
- Case 1: $\exists x, y, \exists x', y'$, s.t. $x \wedge y >_\pi x' \wedge y'$ and
$min(x, y) <_\Pi min(x', y')$, or
$min(x, y) =_\Pi min(x', y')$ and $max(x, y) \leq_\Pi max(x', y')$
Since Pareto-independence is respected $x \wedge y >_\pi x' \wedge y'$
$\Rightarrow x >_\Pi x'$ and $y >_\Pi y'$
$\Rightarrow min(x, x') >_\Pi min(y, y')$ and $max(x, x') >_\Pi max(y, y')$.
Hence contradiction.
- Case 2: $\exists x, y, \exists x', y'$, s.t. $x \wedge y =_\pi x' \wedge y'$ and
$min(x, y) \neq_\Pi min(x', y')$, or $max(x, y) \neq_\Pi max(x', y')$
Since Pareto-independence is respected $x \wedge y =_\pi x' \wedge y'$
$\Rightarrow x =_\Pi x'$ and $y =_\Pi y'$
$\Rightarrow min(x, y) =_\Pi min(x', y')$, and $max(x, y) =_\Pi max(x', y')$
A contradiction again.
- PROOF THAT I_{Pareto} IMPLIES $I_{leximax}$. This proof can be done in the same manner as $I_{Pareto}(X, Z, Y) \Rightarrow I_{leximax}(X, Z, Y)$.

Proof of Proposition 3.12
- PROOF THAT $I_{leximin}$ IMPLIES I_{POS}. Suppose that not i.e. X and Y are leximin but not POS-independent then we can distinguish two possible situations (other cases, are obtained by symmetry):
- Case 1: $\exists x, y, \exists x', y'$ s.t. $x \geq_\Pi x'$ and $x \wedge y <_\pi x' \wedge y$
Since leximin-independence is respected, $x \wedge y <_\pi x' \wedge y$

$\Rightarrow \begin{cases} min(x,y) <_\Pi min(x',y) \text{ or} \\ min(x,y) =_\Pi min(x',y) \text{ and } max(x,y) <_\Pi max(x',y) \end{cases}$
$\Rightarrow x <_\Pi x'$. Hence contradiction.
- Case 2: $\exists x, y, \exists x', y'$ s.t. $x >_\Pi x'$ and $x \wedge y \leq_\pi x' \wedge y$
Since leximin-independence is respected $x \wedge y \leq_\pi x' \wedge y$
$\Rightarrow min(x,y) \leq_\Pi min(x',y)$ and $max(x,y) \leq_\Pi max(x',y)$
$\Rightarrow x \leq_\Pi x'$. Hence contradiction.

Proof of Proposition 3.13

- PROOF THAT I_{POS} IS EQUIVALENT TO $I_{leximin}$ IN THE BINARY CASE. The leximin-independence implies POS-independence in the general case (see Proposition 3.11). Thus, it is enough to prove that POS-independence implies leximin-independence in the binary case.i.e., A and B are POS but not leximin-independent i.e. $\exists a, b, \exists a', b'$ s.t.
(i) $a \wedge b >_\pi a' \wedge b'$ but the relation $a \wedge b >_{leximin} a' \wedge b'$ is false. This may happen in two cases:
- Case 1: $min(a,b) <_\Pi min(a',b') \Rightarrow min(a,b) <_\Pi a'$ and $min(a,b) <_\Pi b'$
Suppose that $a \leq_\Pi b$ then we have $a <_\Pi a'$ and $a <_\Pi b'$
From the POS-independence, $a <_\Pi a'$ implies:
(ii) $a \wedge b <_\pi a' \wedge b$ and (iii) $a \wedge b' <_\pi a' \wedge b'$
From (i),(ii) and (iii) we have $a \wedge b' <_\pi a' \wedge b' <_\pi a \wedge b <_\pi a' \wedge b$
which contradicts $a <_\pi b'$.
- Case 2: $min(a,b) =_\Pi min(a',b')$ and $max(a,b) \leq_\Pi max(a',b')$
Suppose that $a \leq_\Pi b$ then we have $a =_\Pi min(a',b')$ and $b \leq_\Pi max(a',b')$
Suppose now that $a' \leq_\Pi b'$ then we have $a =_\Pi a'$ and $b \leq_\Pi b'$.
From the POS-independence, $b \leq_\Pi b'$ implies:
(ii) $a \wedge b \leq_\pi a \wedge b'$ and (iii) $a' \wedge b \leq_\pi a' \wedge b'$
From (i),(ii) and (iii) we have $a' \wedge b \leq_\pi a' \wedge b' <_\pi a \wedge b \leq_\pi a \wedge b'$
which contradicts $a =_\Pi a'$.
- PROOF THAT I_{POS} IS EQUIVALENT TO $I_{leximax}$ IN THE BINARY CASE. This can be done in the same manner as I_{POS} is equivalent to $I_{leximin}$.

Proof of Proposition 3.14

- PROOF THAT I_{POS} IS EQUIVALENT TO $I_{leximin}$ IF WE HAVE TWO-LEVELS DISTRIBUTIONS. The leximin-independence implies POS-independence in the general case (see Proposition 3.11). Thus, it is enough to prove that POS-independence implies leximin-independence. Suppose that A and B are POS but not leximin-independent i.e. $\exists a, a', \exists b, b'$ s.t. (i) $a \wedge b >_\pi a' \wedge b'$ but the relation $a \wedge b >_{leximin} a' \wedge b'$ is false. This may happen in two cases:

- Case 1: $min(a,b) <_\Pi min(a',b') \Rightarrow min(a,b) <_\Pi a'$ and $min(a,b) <_\Pi b'$
Suppose that $a \leq_\Pi b$ then from Case 1 of the previous proof, we have
$a \wedge b' <_\pi a' \wedge b' <_\pi a \wedge b <_\pi a' \wedge b$ which contradicts the fact that the distributions have only two levels.
- Case 2: $min(a,b) =_\Pi min(a',b')$ and $max(a,b) \leq_\Pi max(a',b')$
Suppose that $a \leq_\Pi b$ then we have $a =_\Pi min(a',b')$ and $b \leq_\Pi max(a',b')$
Suppose now that $a' \leq_\Pi b'$ then we have $a =_\Pi a'$ and $b \leq_\Pi b'$.
From the POS-independence, $a =_\Pi a'$ and $b \leq_\Pi b'$ imply respectively:
$a \wedge b =_\pi a' \wedge b$ and $a \wedge b' =_\pi a' \wedge b'$
$a \wedge b \leq_\pi a \wedge b'$ and $a' \wedge b \leq_\pi a' \wedge b'$
Moreover, from (i) we deduce that $a \wedge b$ is among the top elements (i.e. $\mathbf{Acc}(a \wedge b) = 1$) since we have two-levels distributions. Thus $a \wedge b =_\pi a \wedge b' =_\pi a' \wedge b'$.
Hence contradiction.
- Proof that I_{POS} is equivalent to $I_{leximax}$ if we have two-levels distributions.
This can be done in the same manner as I_{POS} is equivalent to $I_{leximin}$.

Proof of Proposition 3.15 We need to prove that
(a) $\Pi(x \wedge y \mid_m z) = min(\Pi(x \mid_m z), \Pi(y \mid_m z)), \forall xyz$
$\Leftrightarrow$ (b) $\Pi(x \wedge y \wedge z) = min(\Pi(x \wedge z), \Pi(y \wedge z)), \forall xyz.$
Firstly we prove that $(a) \Rightarrow (b)$. Assume that (a) is true, then this means we have two cases:
- Case 1: $\Pi(x \mid_m z) < \Pi(y \mid_m z) \Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z)$
(resp. $\Pi(x \mid_m z) > \Pi(y \mid_m z) \Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(y \mid_m z)$)
$\Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) < 1$ (otherwise if $\Pi(x \mid_m z) = 1$ we will have a contradiction with $\Pi(x \mid_m z) < \Pi(y \mid_m z)$)
$\Rightarrow$ (i) $\Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) = \Pi(x \wedge y \wedge z) = \Pi(x \wedge z) < \Pi(z)$
Moreover, we have by definition $\Pi(y \wedge z) \leq \Pi(z)$, then:

- if (i1) $\Pi(y \wedge z) = \Pi(z)$ then (i) + (i1) $\Rightarrow \Pi(x \wedge z) < \Pi(y \wedge z) = \Pi(z)$
  $\Rightarrow \Pi(x \wedge y \wedge z) = \Pi(x \wedge z) < \Pi(y \wedge z)$,

- if (i2) $\Pi(y \wedge z) < \Pi(z) \Rightarrow \Pi(y \mid_m z) = \Pi(y \wedge z)$
  and we have by assumption $\Pi(x \mid_m z) < \Pi(y \mid_m z)$, thus
  (i) + (i2) $\Rightarrow \Pi(x \wedge z) < \Pi(y \wedge z)$
  $\Rightarrow \Pi(x \wedge y \wedge z) = \Pi(x \wedge z) < \Pi(y \wedge z)$

- Case 2: $\Pi(x \mid_m z) = \Pi(y \mid_m z) \Rightarrow$ (i) $\Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) = \Pi(y \mid_m z)$
Moreover, we have by definition $\Pi(x \wedge y \mid_m z) = 1$ or $\Pi(x \wedge y \wedge z)$, then:

- if (i1) $\Pi(x \wedge y \mid_m z) = 1$ then (i)+(i1) $\Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) = \Pi(y \mid_m z) = 1$
  $\Rightarrow \Pi(x \wedge y \wedge z) = \Pi(x \wedge z) = \Pi(y \wedge z) = \Pi(z)$,

- if (i2) $\Pi(x \wedge y \mid_m z) = \Pi(x \wedge y \wedge z)$
  (i)+(i2) $\Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) = \Pi(y \mid_m z) = \Pi(x \wedge y \wedge z) = \Pi(x \wedge z) = \Pi(y \wedge z) < \Pi(z)$

We now prove that $(b) \Rightarrow (a)$, assume that (b) is true, this again corresponds to two possible cases:

- **Case 1**: $\Pi(x \wedge z) < \Pi(y \wedge z) \Rightarrow$ (i) $\Pi(x \wedge y \wedge z) = \Pi(x \wedge z)$

(resp. $\Pi(x \wedge z) > \Pi(y \wedge z) \Rightarrow \Pi(x \wedge y \wedge z) = \Pi(y \wedge z)$)

Moreover, we have by definition $\Pi(x \wedge z) \leq \Pi(z)$, then:

- if (i1) $\Pi(x \wedge z) = \Pi(z)$ then (i) + (i1) $\Rightarrow \Pi(x \wedge y \wedge z) = \Pi(x \wedge z) = \Pi(z)$
  $\Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) = 1$
  Moreover, $\Pi(y \wedge z) = \Pi(z)$
  (since $\Pi(x \wedge y \wedge z) \leq \Pi(x \wedge z) \leq \Pi(z)$ by definition and $\Pi(x \wedge y \wedge z) = \Pi(z)$ from (i) and (i1))
  $\Rightarrow \Pi(y \mid_m z) = 1$
  $\Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) = \Pi(y \mid_m z) = 1$,

- if (i2) $\Pi(x \wedge z) < \Pi(z)$ then (i) + (i2) $\Rightarrow \Pi(x \wedge y \wedge z) = \Pi(x \wedge z) < \Pi(z)$
  $\Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \wedge y \wedge z)$ and $\Pi(x \mid_m z) = \Pi(x \wedge z)$
  thus $\Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z)$ (since $\Pi(x \wedge y \wedge z) = \Pi(x \wedge z)$)
  Moreover, $\Pi(y \wedge z) \leq \Pi(y \mid_m z)$ (by definition)
  and we have by assumption $\Pi(x \wedge z) < \Pi(y \wedge z)$, thus
  $\Pi(x \wedge z) < \Pi(y \wedge z) \leq \Pi(y \mid_m z)$
  thus $\Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) < \Pi(y \mid_m z)$

- **Case 2**: $\Pi(x \wedge z) = \Pi(y \wedge z) \Rightarrow$ (i) $\Pi(x \wedge y \wedge z) = \Pi(x \wedge z) = \Pi(y \wedge z)$.

Moreover, we have by definition $\Pi(x \wedge y \wedge z) \leq \Pi(z)$, then:

- if (i1) $\Pi(x \wedge y \wedge z) < \Pi(z)$ then (i)+(i1) $\Rightarrow \Pi(x \wedge z) = \Pi(y \wedge z) < \Pi(z)$
  thus $\Pi(x \wedge y \mid_m z) = \Pi(x \wedge y \wedge z)\Pi(x \mid_m z) = \Pi(x \wedge z)$ and
  $\Pi(y \mid_m z) = \Pi(y \wedge z)$
  $\Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) = \Pi(y \mid_m z)$ (from (i))

- if (i2) $\Pi(x \wedge y \wedge z) = \Pi(z)$ then (i)+(i2) $\Rightarrow \Pi(x \wedge z) = \Pi(y \wedge z) = \Pi(z)$
  $\Rightarrow \Pi(x \wedge y \mid_m z) = \Pi(x \mid_m z) = \Pi(y \mid_m z) = 1$.

Proof of Proposition 3.16 We want to prove that I_{MS} implies I_{Prod}. Suppose that $\exists x, \exists y$, such that (i) $\Pi(x \wedge y) \neq \Pi(x) \cdot \Pi(y)$.
Suppose that the distribution on X is uniform (from Proposition 3.19), then $\Pi(x) = 1$, thus $\Pi(x \wedge y) < 1$ (since $\Pi(x \wedge y) < \Pi(y)$). Hence, $\Pi(y \mid x) = \Pi(x \wedge y)$.
Moreover $\Pi(y \mid x) = \Pi(y)$ since X and Y are MS-independent.
Hence contradiction.

Proof of Proposition 3.17 We want to prove that the non-interactivity relation (see (3.14)) can be defined in a purely relational setting. The proof is immediate X and Y are NI-independent means $\Pi(x \wedge y \wedge z) = min(\Pi(x \wedge y), \Pi(x \wedge z))$. Namely, $\Pi(x \wedge y \wedge z) = \Pi(x \wedge y)$ or $\Pi(x \wedge y \wedge z) = \Pi(y \wedge z)$ which is equivalent to: $x \wedge y \wedge z =_\Pi x \wedge y$ or $x \wedge y \wedge z =_\Pi y \wedge z$ (since if $\omega \geq_\pi \omega'$ iff $\pi(\omega) \geq \pi(\omega')$ then $\phi \geq_\Pi \psi'$ iff $\Pi(\phi) \geq \Pi(\psi')$).

Proof of Proposition 3.18 It is obvious that if X and Y are MS-independent then they are Pareto-independent from Propositions 3.10 and the fact that MS-independence implies that one of the local plausibility relations on D_X or D_Y should be uniform.
We now prove that if X and Y are Pareto-independent then they are MS-independent in π.
Suppose that $\exists x, \exists y$, such that $\Pi(x \mid y) \neq \Pi(x)$, then we can distinguish two cases:
- **Case 1**: $\Pi(x) = 1 \Rightarrow \Pi(x \mid y) < 1$

$$\Rightarrow \begin{cases} \Pi(x \wedge y) < \Pi(y) \text{ (conditioning definition) and} \\ \Pi(x \wedge y) < \Pi(x) \text{ (indeed } \Pi(x \mid y) = \Pi(x \wedge y) < 1 = \Pi(x)) \end{cases}$$

$\Rightarrow \exists x', \exists y'$, s.t. $\Pi(x \wedge y) < \Pi(x' \wedge y)$ and $\Pi(x \wedge y) < \Pi(x \wedge y')$
$\Rightarrow \exists x', \exists y'$, s.t. $x \wedge y <_\pi x' \wedge y$ and $x \wedge y <_\pi x \wedge y'$
Since the plausibility relation $\geq_\pi$ is Pareto-decomposable, $\exists x', \exists y'$, s.t. $x <_\Pi x'$ and $y <_\Pi y'$ which contradicts proposition 3.10.
- **Case 2**: $\Pi(x) \neq 1$, then the two possible situations are:

- $\Pi(x \mid y) = 1$
  $\Rightarrow \Pi(x \wedge y) = \Pi(y)$ (conditioning definition)
  $\Rightarrow \forall x', \Pi(x \wedge y) \geq \Pi(x' \wedge y)$.
  $\Rightarrow \forall x', x \wedge y \geq_\pi x' \wedge y$.
  Since the plausibility relation $\geq_\pi$ is Pareto-decomposable, we have $\forall x', x \geq_\Pi x'$ thus x is the top element. However, from $\Pi(x) < 1$ we deduce that x is not the top element. Hence contradiction.

- $\Pi(x \mid y) \neq \Pi(x) < 1$

$$\Rightarrow \begin{cases} \Pi(x \wedge y) < \Pi(y) \text{ (conditioning definition) and} \\ \Pi(x \wedge y) < \Pi(x) \text{ (indeed we have by definition } \Pi(x \wedge y) \leq \Pi(x) \\ \text{ moreover } \Pi(x \mid y) = \Pi(x \wedge y) \neq \Pi(x)) \end{cases}$$

$\Rightarrow \exists x', \exists y'$, s.t. $\Pi(x \wedge y) < \Pi(x' \wedge y)$ and $\Pi(x \wedge y) < \Pi(x \wedge y')$

$\Rightarrow \exists x', \exists y'$, s.t. $x \wedge y <_\pi x' \wedge y$ and $x \wedge y <_\pi x \wedge y'$

Since the plausibility relation $\geq_\pi$ is Pareto-decomposable, $\exists x', \exists y'$, s.t. $x <_\Pi x'$ and $y <_\Pi y'$ which contradicts proposition 3.10.

Proof of Proposition 3.19 We want to prove that X and Y are MS-independent in a possibility distribution π if and only if they are Pareto-independent in its associated plausibility relation $\geq_\pi$. We firstly prove that given a plausibility relation π, then if X and Y are Pareto-independent in its associated plausibility relation $\geq_\pi$ then they are MS-independent in π. Suppose that $\exists x, \exists y$, such that $\Pi(x \mid y) \neq \Pi(x)$. Since $\Pi(x) \in [0, 1]$ we can distinguish two cases:

- **Case 1**: $\Pi(x) = 1 \Rightarrow \Pi(x \mid y) < 1$

$$\Rightarrow \begin{cases} \Pi(x \wedge y) < \Pi(y) \text{ (conditioning definition) and} \\ \Pi(x \wedge y) < \Pi(x) \text{ (indeed } \Pi(x \mid y) = \Pi(x \wedge y) < 1 = \Pi(x)) \end{cases}$$

$$\Rightarrow \exists x', \exists y', \text{ s.t. } \begin{cases} \Pi(x \wedge y) < \Pi(x' \wedge y) \text{ and} \\ \Pi(x \wedge y) < \Pi(x \wedge y') \end{cases}$$

$$\Rightarrow \exists x', \exists y', \text{ s.t. } \begin{cases} x \wedge y <_\pi x' \wedge y \text{ and} \\ x \wedge y <_\pi x \wedge y' \end{cases}$$

Since the plausibility relation $\geq_\pi$ is Pareto-decomposable, $\exists x', \exists y'$, s.t. $x <_\Pi x'$ and $y <_\Pi y'$ which contradicts proposition 3.10.

- **Case 2**: $\Pi(x) \neq 1$, then the two possible situations are:

- $\Pi(x \mid y) = 1$

  $\Rightarrow \Pi(x \wedge y) = \Pi(y)$ (conditioning definition)

  $\Rightarrow \forall x', \Pi(x \wedge y) \geq \Pi(x' \wedge y)$.

  $\Rightarrow \forall x', x \wedge y \geq_\pi x' \wedge y$.

  Since the plausibility relation $\geq_\pi$ is Pareto-decomposable, we have $\forall x', x \geq_\Pi x'$ thus x is the top element. However, from $\Pi(x) < 1$ we deduce that x is not the top element. Hence contradiction.

- $\Pi(x \mid y) \neq \Pi(x) < 1$

$$\Rightarrow \begin{cases} \Pi(x \wedge y) < \Pi(y) \text{ (conditioning definition) and} \\ \Pi(x \wedge y) < \Pi(x) \text{ (indeed we have by definition } \Pi(x \wedge y) \leq \Pi(x) \\ \text{ moreover } \Pi(x \mid y) = \Pi(x \wedge y) \neq \Pi(x)) \end{cases}$$

$$\Rightarrow \exists x', \exists y', \text{ s.t. } \begin{cases} \Pi(x \wedge y) < \Pi(x' \wedge y) \text{ and} \\ \Pi(x \wedge y) < \Pi(x \wedge y') \end{cases}$$

$$\Rightarrow \exists x', \exists y', \text{ s.t. } \begin{cases} x \wedge y <_\pi x' \wedge y \text{ and} \\ x \wedge y <_\pi x \wedge y' \end{cases}$$

Since the plausibility relation $\geq_\pi$ is Pareto-decomposable, $\exists x', \exists y'$, s.t. $x <_\Pi x'$ and $y <_\Pi y'$ which contradicts proposition 3.10.

We now prove that if X and Y are MS-independent in a possibility distribution π then they are Pareto-independent in its associated plausibility relation $\geq_\pi$. Suppose that

(a) $\exists x, y, \exists x', y'$ s.t. $x \wedge y \geq_\pi x' \wedge y'$ but $x <_\Pi x'$. Since MS is respected, we have:

$$\begin{cases} \Pi(x \mid y) = \Pi(x) \text{ and} \\ \Pi(x \mid y') = \Pi(x) \end{cases}$$

Moreover, from (a) we deduce that x is not the top element i.e. $\Pi(x) < 1$, thus:

$$\begin{cases} \text{(i) } \Pi(x \mid y) = \Pi(x) < 1 \text{ and} \\ \text{(ii) } \Pi(x \mid y') = \Pi(x) < 1 \end{cases}$$

Moreover:

- $\Pi(x \wedge y) < \Pi(y)$, indeed we have by definition $\Pi(x \wedge y) \leq \Pi(y)$ but if $\Pi(x \wedge y) = \Pi(y)$ then $\Pi(x \mid y) = 1$ which contradicts (i).
- $\Pi(x \wedge y') < \Pi(y')$, indeed we have by definition $\Pi(x \wedge y') \leq \Pi(y')$ but if $\Pi(x \wedge y') = \Pi(y')$ then $\Pi(x \mid y') = 1$ which contradicts (ii).

Thus (b): $\begin{cases} \Pi(x \wedge y) = \Pi(x) < 1 \text{ and } \Pi(x \wedge y) < \Pi(y) \text{ and} \\ \Pi(x \wedge y') = \Pi(x) < 1 \text{ and } \Pi(x \wedge y') < \Pi(y') \end{cases}$

(a) + (b) $\Rightarrow$ (c): $min(\Pi(y), \Pi(y')) > \Pi(x \wedge y') = \Pi(x \wedge y) = \Pi(x) \geq \Pi(x' \wedge y')$.

Moreover we have $\Pi(x' \mid y') = \Pi(x')$ (From MS) and $\Pi(x' \wedge y') < \Pi(y')$ (From (c)). Then we can deduce that $\Pi(x' \wedge y') = \Pi(x')$. If we use this relation in (c) we obtain $\Pi(x) \geq \Pi(x') \Rightarrow x \geq_\Pi x'$ which contradicts (a).

Proof of Proposition 3.20 We want to prove that if X and Y are M-independent in a

possibility distribution π, then they are PO-independent in the plausibility relation induced by π. Suppose that X and Y are not PO-independent then, we can distinguish two cases:

- $\exists x, x', \exists y$ s.t. $x >_\Pi x'$ but $x \wedge y \leq_\pi x' \wedge y$
  $\Rightarrow$ (i) $\Pi(x) > \Pi(x')$ but (ii) $\Pi(x \wedge y) \leq \Pi(x' \wedge y)$.

  Since M-independence is respected, we have $\Pi(x \wedge y) = \Pi(x)$ and $\Pi(x' \wedge y) = \Pi(x')$. When using these two relations in (ii) we obtain $\Pi(x) \leq \Pi(x')$ which contradicts (i).

- $\exists x, x', \exists y$ s.t. $x =_\Pi x'$ but $x \wedge y >_\pi x' \wedge y$ (or $x \wedge y <_\pi x' \wedge y$)
  $\Rightarrow$ (i) $\Pi(x) = \Pi(x')$ but (ii) $\Pi(x \wedge y) > \Pi(x' \wedge y)$ (or $\Pi(x \wedge y) < \Pi(x' \wedge y)$).

  Since M-independence is respected, we have $\Pi(x \wedge y) = \Pi(x)$ and $\Pi(x' \wedge y) = \Pi(x')$. When using these two relations in (ii) we obtain $\Pi(x) < \Pi(x')$ which contradicts (i).

Proof of Proposition 3.21 We want to prove that if X and Y are Prod-independent in a strictly positive possibility distribution π, then they are POS-independent in the plausibility relation induced by π. Suppose that X and Y are not POS-independent then, the possible situations are:

- $\exists x, x', \exists y$ s.t. $x >_\Pi x'$ but $x \wedge y \leq_\pi x' \wedge y$
  $\Rightarrow$ (i) $\Pi(x) > \Pi(x')$ but (ii) $\Pi(x \wedge y) \leq \Pi(x' \wedge y)$.

  Since Prod-independence is respected, we have:
  $\Pi(x \wedge y) = \Pi(x) \cdot \Pi(y)$, and $\Pi(x' \wedge y) = \Pi(x') \cdot \Pi(y)$.
  When using these two relations in (ii) we obtain:
  $\Pi(x) \cdot \Pi(y) \leq \Pi(x') \cdot \Pi(y)$
  $\Rightarrow \Pi(x) \leq \Pi(x')$ which contradicts (i).

- $\exists x, x', \exists y$ s.t. $x =_\Pi x'$ but $x \wedge y >_\pi x' \wedge y$ (or $x \wedge y <_\pi x' \wedge y$)
  $\Rightarrow$ (i) $\Pi(x) = \Pi(x')$ but (ii) $\Pi(x \wedge y) > \Pi(x' \wedge y)$ (or $\Pi(x \wedge y) < \Pi(x' \wedge y)$)

  Since Prod-independence is respected, we have:
  $\Pi(x \wedge y) = \Pi(x) \cdot \Pi(y)$, and $\Pi(x' \wedge y) = \Pi(x') \cdot \Pi(y)$.
  When using these two relations in (ii) we obtain:
  $\Pi(x) \cdot \Pi(y) < \Pi(x') \cdot \Pi(y)$
  $\Rightarrow \Pi(x) < \Pi(x')$ which contradicts (i).

Proof of Proposition 3.22 We want to prove that if X and Y are NI-independent in a strictly positive possibility distribution π, then they are PT-independent in the plausibility relation induced by π.
Suppose X and Y are NI-independent but not PT-independent. More formally, $\exists x, \exists y$ s.t. (i)

$\mathbf{Acc}(x \wedge y) \neq min(\mathbf{Acc}(x), \mathbf{Acc}(y))$.
Hence $\mathbf{Acc}(x \wedge y) = -1, Acc(x) = 0, Acc(y) = 0$ (from Proposition 2.1 item 3). Hence $\pi(x \wedge y) < 1, \Pi(x) = 1, \Pi(y) = 1$. This is impossible since NI-independence implies that $\forall x \in D_X, \forall y \in D_Y, \pi(x \wedge y) = min(\Pi(x), \pi(y))$.

B.2 Proofs for graphoid properties

Proof of Proposition 3.23
- Decomposition property for I_{PO}.
We want to prove that $I_{PO}(X, \emptyset, Y \cup W) \Rightarrow I_{PO}(X, \emptyset, Y)$ and $I_{PO}(X, \emptyset, W)$.
By symmetry we only prove that $I_{PO}(X, \emptyset, Y \cup W) \Rightarrow I_{PO}(X, \emptyset, Y)$.
Thus, we need to prove that:
if (i) $\forall y \in D_Y, \forall w \in D_W, \forall x_i : x_j \in D_X, x_i >_\Pi x_j$ iff $x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$
then (ii) $\forall y \in D_Y : \forall x_i, x_j \in D_X, x_i >_\Pi x_j$ iff $x_i \wedge y >_\Pi x_j \wedge y$.

Let us consider two instances $x_i, x_j \in D_X$ s.t. $x_i >_\Pi x_j$ (resp. $x_i =_\Pi x_j$)
This implies that $\forall y \in D_Y, \forall w \in D_W$, $x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$ (resp. $x_i \wedge y \wedge w =_\pi x_j \wedge y \wedge w$) (from (i))
$\Rightarrow \forall y \in D_Y, max_w x_i \wedge y \wedge w >_\pi max_w x_j \wedge y \wedge w$ (resp. $max_w x_i \wedge y \wedge w =_\pi max_w x_j \wedge y \wedge w$)
$\Rightarrow \forall y \in D_Y, x_i \wedge y >_\Pi x_j \wedge y$ (resp. $x_i \wedge y =_\Pi x_j \wedge y$).

- Weak union property for I_{PO}.
We want to prove that $I_{PO}(X, \emptyset, Y \cup W) \Rightarrow I_{PO}(X, Y, W)$.
Thus, we need to prove that:
if (i) $\forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i >_\Pi x_j$ iff $x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$
then (ii) $\forall y \in D_Y, \forall w \in D_W$: $\forall x_i, x_j \in D_X, x_i \wedge y >_\Pi x_j \wedge y$ iff $x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$.

Let $x_i, x_j \in D_X$, $y' \in D_Y$ and $w' \in D_W$ s.t.
$x_i \wedge y' \wedge w' >_\pi x_j \wedge y' \wedge w'$ (resp. $x_i \wedge y' \wedge w' =_\pi x_j \wedge y' \wedge w'$)
This implies that $x_i >_\Pi x_j$ (resp. $x_i =_\Pi x_j$) (from (i))
$\Rightarrow \forall y \in D_Y, \forall w \in D_W, x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$ (resp. $x_i \wedge y \wedge w =_\pi x_j \wedge y \wedge w$) (from (i))
$\Rightarrow \forall y \in D_Y, max_w x_i \wedge y \wedge w >_\pi max_w x_j \wedge y \wedge w$ (resp. $max_w x_i \wedge y \wedge w =_\pi max_w x_j \wedge y \wedge w$)
$\Rightarrow \forall y \in D_Y, x_i \wedge y >_\Pi x_j \wedge y$ (resp. $x_i \wedge y =_\Pi x_j \wedge y$)
$\Rightarrow x_i \wedge y' >_\Pi x_j \wedge y'$ (resp. $x_i \wedge y' =_\Pi x_j \wedge y'$) (when Y takes the particular instance y')

- Contraction property for I_{PO}.

We want to prove that $I_{PO}(X,Y,W)$ and $I_{PO}(X,\emptyset,Y) \Rightarrow I_{PO}(X,\emptyset,Y \cup W)$.
Thus, we need to prove that:
if (i) $\forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i \wedge y >_\Pi x_j \wedge y$ iff $x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$ and (ii)
$\forall y \in D_Y, \forall x_i, x_j \in D_X : x_i >_\Pi x_j$ iff $x_i \wedge y >_\Pi x_j \wedge y$
then (iii) $\forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i >_\Pi x_j$ iff $x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$.

Let us consider two instances $x_i, x_j \in D_X$ s.t. $x_i >_\Pi x_j$ (resp. $x_i =_\Pi x_j$)
This implies that $\forall y \in D_Y, x_i \wedge y >_\Pi x_j \wedge y$ (resp. $x_i \wedge y =_\Pi x_j \wedge y$) (from (ii))
$\Rightarrow \forall y \in D_Y, \forall w \in D_W, x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$ (resp. $x_i \wedge y \wedge w =_\pi x_j \wedge y \wedge w$) (from (i)).

- Intersection property for I_{PO}.
We want to prove that $I_{PO}(X,W,Y)$ and $I_{PO}(X,Y,W) \Rightarrow I_{PO}(X,\emptyset,Y \cup W)$.
Thus we need to prove that:
if (i) $\forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$ iff $x_i \wedge w >_\Pi x_j \wedge w$ and (ii)
$\forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$ iff $x_i \wedge y >_\Pi x_j \wedge y$
then (ii) $\forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w$ iff $x_i >_\Pi x_j$

Suppose that $\exists x', x'' \in D_X, y' \in D_Y, w' \in D_W$ s.t. $x' >_\Pi x''$ (resp. $x' =_\Pi x''$) while
$x'' \wedge y' \wedge w' >_\pi x' \wedge y' \wedge w'$ (resp. $x' \wedge y' \wedge w' \neq_\pi x'' \wedge y' \wedge w'$)
$\Rightarrow x'' \wedge y' >_\Pi x' \wedge y'$ (resp. $x' \wedge y' \neq_\pi x'' \wedge y'$) (from (ii))
$\Rightarrow \forall w \in D_W, x'' \wedge y' \wedge w >_\pi x' \wedge y' \wedge w$ (resp. $x' \wedge y' \wedge w \neq_\pi x'' \wedge y' \wedge w$) (from (ii))
$\Rightarrow \forall w \in D_W, x'' \wedge w >_\Pi x' \wedge w$ (resp. $x' \wedge w \neq_\Pi x'' \wedge w$) (from (i))
$\Rightarrow max_w x'' \wedge w >_\Pi max_w x' \wedge w$ (resp. $max_w x' \wedge w \neq_\Pi max_w x'' \wedge w$)
$\Rightarrow x'' >_\Pi x'$ (resp. $x' \neq_\Pi x''$)
Hence contradiction.

Proof of Proposition 3.24

Decomposition property for I_{POS}. We want to prove that
$I_{POS}(X,\emptyset,Y \cup W) \Rightarrow I_{POS}(X,\emptyset,Y)$. Thus, we need to prove that if
$\begin{cases} (i_1) \forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i >_\Pi x_j \text{ iff } x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w \text{ and} \\ (i_2) \forall x \in D_X : \forall y_k, y_l \in D_Y, \forall w_m, w_n \in D_W, y_k \wedge w_m >_\Pi y_l \wedge w_n \text{ iff } y_k \wedge w_m \wedge x >_\pi y_l \wedge w_n \wedge x \end{cases}$
then $\begin{cases} \forall y \in D_Y : \forall x_i, x_j \in D_X, x_i >_\pi x_j \text{ iff } x_i \wedge y >_\pi x_j \wedge y, \text{ and} \\ \forall x \in D_X : \forall y_k, y_l \in D_Y, y_k >_\Pi y_l \text{ iff } y_k \wedge x >_\pi y_l \wedge x \end{cases}$
Suppose that
$\exists x', x'', \exists y'$ s.t. (a) $x' >_\pi x''$ and (b) $x' \wedge y' \leq_\pi x'' \wedge y'$. Then, we have $\forall y \in D_Y, \forall w \in D_W, x' \wedge y \wedge w >_\pi x'' \wedge y \wedge w$(From (a) and (i_1))

$\Rightarrow \forall y \in D_Y, x' \wedge y >_\Pi x'' \wedge y$ (By maximization on W).
Hence, contradiction with (b) since y' can take the particular value y.
Suppose now that $\exists y', y'', \exists x'$ s.t. (a) $y' >_\pi y''$ and (b) $y' \wedge x' \leq_\pi y'' \wedge x'$.
From (a) we can deduce that it exists a particular value w' of W such that
$\forall w \in D_W, y' \wedge w' >_\pi y'' \wedge w$, thus $\forall x \in D_X, \forall w \in D_W, y' \wedge w' \wedge x >_\pi y'' \wedge w \wedge x$ (From (i_2))
$\Rightarrow y' \wedge x >_\Pi y'' \wedge x$ (since $\forall x \in D_X, y' \wedge w'$ is more plausible than $y'' \wedge w$ for any value w of W including the particular value w'). Hence contradiction with (b).
The proof for $I_{POS}(X, Z, W)$ is analogous.

Weak union property for I_{POS}. We want to prove that $I_{POS}(X, \emptyset, Y \cup W) \Rightarrow I_{POS}(X, Y, W)$.
Thus, we need to prove that if:
$\begin{cases} (i_1) \forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i >_\Pi x_j \text{ iff } x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w \text{ and} \\ (i_2) \forall x \in D_X : \forall y_k, y_l \in D_Y, \forall w_m, w_n \in D_W, y_k \wedge w_m >_\Pi y_l \wedge w_n \text{ iff } y_k \wedge w_m \wedge x >_\pi y_l \wedge w_n \wedge x \end{cases}$
then: $\begin{cases} \forall y \in D_Y, \forall w \in D_W : \forall x_i, x_j \in D_X, x_i \wedge y >_\pi x_j \wedge y \text{ iff } x_i \wedge y \wedge w >_\pi x_j \wedge y \wedge w \text{ and} \\ \forall x \in D_X, \forall y \in D_Y : \forall w_m, w_n \in D_W, w_m \wedge y >_\Pi w_n \wedge y \text{ iff } w_m \wedge x \wedge y >_\pi w_n \wedge x \wedge y \end{cases}$
Suppose that $\exists x', x'', \exists y', \exists w'$ s.t.
(a) $x' \wedge y' \wedge w' >_\pi x'' \wedge y' \wedge w'$ and (b) $x' \wedge y' \leq_\pi x'' \wedge y'$.
Then, we have $x' >_\Pi x''$ (From (a) and (i_1))
$\Rightarrow \forall w \in D_W, x' \wedge y' \wedge w >_\pi x'' \wedge y' \wedge w$ (From (i_1))
$\Rightarrow x' \wedge y' >_\pi x'' \wedge y'$ ((By maximization on W).) which contradicts (b).
Suppose now that $\exists w', w'', \exists x', \exists y'$, s.t.
(a) $w' \wedge y' >_\pi w'' \wedge y'$ and (b) $w' \wedge x' \wedge y' \leq_\pi w'' \wedge x' \wedge y'$
then, we have $y' \wedge w' >_\Pi y' \wedge w''$ (From (a))
$\Rightarrow y' \wedge w' \wedge x' >_\pi y' \wedge w'' \wedge x'$(From (i_2))
$\Rightarrow x' \wedge y' \wedge w' >_\pi x' \wedge y' \wedge w''$
Moreover $x' \wedge y' \wedge w' \leq \_\pi x' \wedge y' \wedge w''$ (From (b)), hence contradiction.

Proof of Proposition 3.25
- Decomposition property for $I_{leximax}$.
We want to prove that $I_{leximax}(X, \emptyset, Y \cup W) \Rightarrow I_{leximax}(X, \emptyset, Y)$ and $I_{leximax}(X, \emptyset, W)$.
By symmetry we only prove that if (i) $I_{leximax}(X, \emptyset, Y \cup W)$ is true then (ii) $I_{leximax}(X, \emptyset, Y)$ is true.
Suppose that $I_{leximax}(X, \emptyset, Y \cup W)$ is true but not $I_{leximax}(X, \emptyset, Y)$.
Let us consider the two cases where $I_{leximax}(X, \emptyset, Y)$ is falsified:

Case 1: $\exists x, x' \in D_X, \exists y, y' \in D_Y$ s.t. (a) $x \wedge y >_\Pi x' \wedge y'$ but

(i1) $max(x,y) <_\Pi max(x',y')$ or
(i2) $max(x,y) =_\Pi max(x',y')$ and $min(x,y) \leq_\Pi min(x',y')$

By definition we have $x \wedge y =_\Pi max_w x \wedge y \wedge w$ and $x' \wedge y' =_\Pi max_w x' \wedge y' \wedge w$
Let w_i be one of the instances of W which maximizes $x \wedge y$ and w_j be one of the instances of W which maximizes $x' \wedge y'$, then:
$x \wedge y =_\Pi x \wedge y \wedge w_i$ and $x' \wedge y' =_\Pi x' \wedge y' \wedge w_j$

From (a) we have $x \wedge y \wedge w_i >_\pi x' \wedge y' \wedge w_j$ then from (i) this relation implies:
(ii1) $max(x, y \wedge w_i) >_\Pi max(x', y' \wedge w_j)$ or
(ii2) $max(x, y \wedge w_i) =_\Pi max(x', y' \wedge w_j)$ and $min(x, y \wedge w_i) >_\Pi min(x', y' \wedge w_j)$

Then it is enough to show that $y \wedge w_i =_\Pi y$ and $y' \wedge w_j =_\Pi y'$ in order to prove that (i1) and (i2) contradict (ii1) and (ii2).

Let us prove that $y \wedge w_i =_\Pi y$ (the proof of $y' \wedge w_j =_\Pi y'$ is analogous).
By definition we have : (b) $y =_\Pi max_w y \wedge w =_\Pi max(y \wedge w_i, max_{w'_i \neq w_i} y \wedge w'_i)$
Moreover w_i maximizes $x \wedge y$ then $\forall w'_i \in D_W$ s.t. $w'_i \neq_\Pi w_i$:
$x \wedge y \wedge w_i \geq_\pi x \wedge y \wedge w'_i$. Then,

- if $x \wedge y \wedge w_i >_\pi x \wedge y \wedge w'_i$, then from (i), we can distinguish two cases:
  - $max(x, y \wedge w_i) >_\Pi max(x, y \wedge w'_i)$
    $\Rightarrow max(x, y \wedge w_i) >_\Pi x$ and $max(x, y \wedge w_i) >_\Pi y \wedge w'_i$
    $\Rightarrow y \wedge w_i >_\Pi x$ (otherwise $x >_\Pi x$)
    $\Rightarrow y \wedge w_i >_\Pi y \wedge w'_i$
  - $max(x, y \wedge w_i) =_\Pi max(x, y \wedge w'_i)$ and $min(x, y \wedge w_i) >_\Pi min(x, y \wedge w'_i)$
    $\Rightarrow min(x, y \wedge w'_i) <_\Pi x$ and $min(x, y \wedge w'_i) <_\Pi y \wedge w_i$
    $\Rightarrow y \wedge w'_i <_\Pi x$ (otherwise $x <_\Pi x$)
    $\Rightarrow y \wedge w_i >_\Pi y \wedge w'_i$
- if $x \wedge y \wedge w_i =_\pi x' \wedge y' \wedge w'_i$, then from (i) we deduce that:
  $max(x, y \wedge w_i) =_\Pi max(x, y \wedge w'_i)$ and $min(x, y \wedge w_i) =_\Pi min(x, y \wedge w'_i)$
  $\Rightarrow y \wedge w_i =_\Pi y \wedge w'_i$

Thus, it is clear that $\forall w'_i \neq w_i, y \wedge w_i \geq_\Pi y \wedge w'_i$, so from (b) we deduce that $y =_\Pi y \wedge w_i$.

Case 2: $\exists x, x' \in D_X, \exists y, y' \in D_Y$ s.t. (b) $x \wedge y =_\Pi x' \wedge y'$ but
(i1) $max(x, y) \neq_\Pi max(x', y')$ or (i2) $min(x, y) \neq_\Pi min(x', y')$

From (b) we have $x \wedge y \wedge w_i =_\pi x' \wedge y' \wedge w_j$ where w_i is one of the instances of W which maximizes $x \wedge y$ and w_j is one of the instances of W which maximizes $x' \wedge y'$.
From (i), $x \wedge y \wedge w_i =_\pi x' \wedge y' \wedge w_j$ implies:
(ii1) $max(x, y \wedge w_i) =_\Pi max(x', y' \wedge w_j)$ and (ii2) $min(x, y \wedge w_i) =_\Pi min(x', y' \wedge w_j)$.
Moreover, we have shown above that $y \wedge w_i =_\Pi y$ and that $y' \wedge w_j =_\Pi y'$ then (i1) and (i2) contradict (ii1) and (ii2).

Appendix C

Proofs of Chapter 5

C.1 Proofs relative to product-based possibilistic networks

Proof of Proposition 5.1 Let π_p be the global joint possibility distribution of ΠG_p computed using (5.1). Let $\Pi(a \mid u_A)$ be the conditional distribution given by the expert on the node A and $\Pi_p(a \mid u_A)$ be the conditional possibility computed from π_p. We want to prove that, $\Pi_p(a \mid u_A) = \Pi(a \mid u_A)$. For any node $A \in V$, let:
- u_A be a possible instantiation of the parent set U_A,
- x_A be a possible instantiation of the descendants set X_A,
- y_A be a possible instantiation of the children set Y_A,
- z_A be a possible instantiation of the non-descendants set Z_A.

Note that for each $T \in V \setminus (X_A \cup A)$, we satisfy $U_T \cap (X_A \cup A) = \emptyset$. Indeed, if there is $Z \in U_T \cap (X_A \cup A)$, then this simply means that the variable T is also a descendant of A and hence contradicts the fact that $T \in V \setminus X_A$. Then using the product-based chain rule (5.1), we have:
$\Pi_p(a \wedge u_A) = max_{x_A, z_A}\{\pi_p(a \wedge z_A \wedge x_A \wedge u_A)\}$
$= max_{x_A, z_A}\{\Pi(a \mid u_A) \cdot \prod\{\Pi(b \mid u_B) : b \in u_A\} \cdot \prod\{\Pi(c \mid u_C) : c \in z_A\} \cdot \prod\{\Pi(e \mid u_E) :$
$e \in x_A, u_E \subseteq a \wedge x_A\}\}$ (Note that $u_B \cap (x_A \wedge a) = \emptyset$ due to the above remark)
$= \Pi(a \mid u_A) \cdot max_{z_A}\{\prod\{\Pi(b \mid u_B) : b \in u_A\} \cdot \prod\{\Pi(c \mid u_C) : c \in z_A\}\} \cdot max_{x_A}\{\prod\{\Pi(e \mid u_E) :$
$e \in x_A, u_E \subseteq a \wedge x_A\}\}$

From Lemma 5.1, we have $max_{x_A}\{\prod\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq a \wedge x_A\}\} = 1$
(since $(a \wedge x_A) \in ((a \wedge u_A \wedge z_A) \wedge x_A))$, then:
(i) $\Pi_p(a \wedge u_A) = \Pi(a \mid u_A) \cdot max_{z_A}\{\prod\{\Pi(b \mid u_B) : b \in u_A\} \cdot \prod\{\Pi(c \mid u_C) : c \in z_A\}\}$
$= \Pi(a \mid u_A) \cdot max_{z_A}\{\prod\{\Pi(b \mid u_B) : b \in u_A\} \cdot \prod\{\Pi(c \mid u_C) : c \in z_A\}\} \cdot max(\Pi(a \mid u_A),$

$\Pi(\neg a \mid u_A))$ (since $max(\Pi(a \mid u_A), \Pi(\neg a \mid u_A)) = 1$)
$= \Pi(a \mid u_A) \cdot max[$
$\Pi(a \mid u_A) \cdot max_{z_A}\{\prod\{\Pi(b \mid u_B) : b \in u_A\} \cdot \prod\{\Pi(c \mid u_C) : c \in z_A\}\},$
$\Pi(\neg a \mid u_A) \cdot max_{z_A}\{\prod\{\Pi(b \mid u_B) : b \in u_A\} \cdot \prod\{\Pi(c \mid u_C) : c \in z_A\}\}]$
$= \Pi(a \mid u_A) \cdot max[$
$\Pi(a \mid u_A) \cdot max_{z_A}\{\prod\{\Pi(b \mid u_B) : b \in u_A\} \cdot \prod\{\Pi(c \mid u_C) : c \in z_A\}\} \cdot$
$max_{x_A}\{\prod\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq a \wedge x_A\}\},$
$\Pi(\neg a \mid u_A) \cdot max_{z_A}\{\prod\{\Pi(b \mid u_B) : b \in u_A\} \cdot \prod\{\Pi(c \mid u_C) : c \in z_A\}\} \cdot$
$max_{x_A}\{\prod\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq \neg a \wedge x_A\}\}]$
(Indeed from Lemma 5.1, we have $max_{x_A}\{\prod\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq a \wedge x_A\}\} = 1$, and $max_{x_A}\{\prod\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq \neg a \wedge x_A\}\} = 1$)
$= \Pi(a \mid u_A) \cdot max[\Pi_p(a \wedge u_A), \Pi_p(\neg a \wedge u_A)]$ (From (i))
$= \Pi(a \mid u_A) \cdot \Pi_p(u_A)$

Then $\Pi(a \mid u_A) = \frac{\Pi_p(a \wedge u_A)}{\Pi_p(u_A)} = \Pi_p(a \mid u_A)$ since we have by definition:
$\Pi_p(a \mid u_A) = \frac{\Pi_p(a \wedge u_A)}{\Pi_p(u_A)}$.

C.2 Proofs relative to min-based possibilistic networks

Proof of Proposition 5.4 Let π_m be the global joint possibility distribution of ΠG_m computed using (5.6). Let $\Pi(a \mid u_A)$ be the conditional distribution given by the expert on the node A and $\Pi_m(a \mid u_A)$ be the conditional possibility computed from π_m. We want to prove that:
either $\Pi_m(a \mid u_A) = \Pi(a \mid u_A)$ or $\Pi_m(a \mid u_A) = 1$.
Moreover if $\Pi_m(a \mid u_A) = 1 \neq \Pi(a \mid u_A)$, then $\Pi(a \mid u_A) > \Pi_m(u_A)$
Let:
- a be a fixed instance of A, and $\neg a = D_A - \{a\}$
- u_A be a possible instantiation of the parent set U_A,
- x_A be a possible instantiation of the descendants set X_A,
- z_A be a possible instantiation of the non-descendants set Z_A,
- $\alpha_1 = \Pi(a \mid u_A)$,
- $\alpha_2 = \Pi(\neg a \mid u_A)$,
- $\beta_1 = min\{\Pi(b \mid u_B) : b \in u_A, u_B \subseteq z_A\}$,
- $\gamma_1 = min\{\Pi(c \mid u_C) : c \in z_A\}$,
- $\delta_1 = min\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq a \wedge x_A \wedge u_A \wedge z_A\}$,
- $\delta_2 = min\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq \neg a \wedge x_A \wedge u_A \wedge z_A\}$.

Then using the min-based chain rule (5.6), we have:
$\Pi_m(a \wedge u_A) = \max_{x_A, z_A} \pi_m(a \wedge u_A \wedge z_A \wedge x_A)$
$= \max_{x_A, z_A} min(\alpha_1, \beta_1, \gamma_1, \delta_1)$ (from 5.6)
$= \max_{z_A} min(\alpha_1, \beta_1, \gamma_1, \max_{x_A} \delta_1)$
$=$ **(i)** $min[\alpha_1, \max_{z_A} min(\beta_1, \gamma_1, \max_{x_A} \delta_1)]$
$= min[\alpha_1, \max_{z_A} min(\beta_1, \gamma_1)]$ (from Lemma 5.2)
$= min[\alpha_1, Max\{min[\alpha_1, \max_{z_A} min(\beta_1, \gamma_1)], min[\alpha_2, \max_{z_A} min(\beta_1, \gamma_1)]\}]$
(since $max(\alpha_1, \alpha_2) = 1$ from the normalization condition)
$= min[\alpha_1, Max\{min[\alpha_1, \max_{z_A} min(\beta_1, \gamma_1, \max_{x_A} \delta_1)], min[\alpha_2, \max_{z_A} min(\beta_1, \gamma_1, \max_{x_A} \delta_2)]\}]$
(from Lemma 5.2)
$= min[\alpha_1, Max\{\Pi_m(a \wedge u_A), \Pi_m(\neg a \wedge u_A)\}]$ (from (i))
$= min[\alpha_1, \Pi_m(u_A)]$ (by definition)
$= min[\Pi(a \mid u_A), \Pi_m(u_A)]$ (by recovering the value of α_1)
Then, we can distinguish two cases:

- $\Pi(a \mid u_A) = 1 \Rightarrow \Pi_m(a \wedge u_A) = \Pi_m(u_A)$
  hence using the definition of conditioning, we get $\Pi_m(a \mid u_A) = 1$
- $\Pi(a \mid u_A) = \alpha < 1 \Rightarrow \Pi_m(a \wedge u_A) = min(\alpha, \Pi_m(u_A))$ Then,
  - if $\alpha \leq \Pi_m(u_A)$ then using the definition of conditioning, we get $\Pi_m(a \mid u_A) = \alpha$
  - if $\alpha > \Pi_m(u_A)$ then $\Pi_m(a \wedge u_A) = \Pi_m(u_A)$
    hence using the definition of conditioning, we get $\Pi_m(a \mid u_A) = 1$

Note that the only case where the conditional value is not recovered is where $\alpha > \Pi_m(u_A)$.

Proof of Proposition 5.5 Let $A \in V$ be a variable in ΠG_m s.t. $\Pi_m(a \mid u_A) \neq \Pi(a \mid u_A)$. Let π'_m be a new joint distribution obtained from ΠG_m by only substituting the value $\Pi(a \mid u_A)$ by the degree 1. We want to prove that $\pi'_m = \pi_m$.

Let $V' \subseteq V$ s.t. $\forall A_j \in V'$, $\Pi(a_j \mid u_{A_j}) < 1$ and **(i)** $\Pi(a_j \mid u_{A_j}) > \Pi_m(u_{A_j})$. Then it is enough to show that for any variable $A_j \in V'$: $\pi_m(v) = \min_{i=1, i \neq j}^{N} \Pi(a_i \mid u_{A_i})$.

Let $d = (A_1, ..., A_N)$ be an ordering of the variables in V such that $\forall A_i, U_{A_i} \subseteq \{A_{i+1}, ..., A_N\}$ and let A_j be any node in V'. We first prove that:
(ii) $\Pi_m(a_{j+1}, ..., a_N) = \min_{i=j+1..N} \Pi(a_i \mid u_{A_i})$
Indeed, we have by definition:
$\Pi_m(a_{j+1}, ..., a_N) = \max_{a_1..a_j} \min_{i=1..N} \Pi(a_i \mid u_{A_i})$

$= \max_{a_1..a_j} min(\min_{i=1..j} \Pi(a_i \mid u_{A_i}), \min_{k=j+1..N} \Pi(a_k \mid u_{A_k}))$
$= min(\max_{a_1..a_j} \min_{i=1..j} \Pi(a_i \mid u_{A_i}), \min_{k=j+1..N} \Pi(a_k \mid u_{A_k}))$
$= \min_{i=j+1..N} \Pi(a_i \mid u_{A_i})$ (since $\max_{a_1..a_j} \min_{i=1..j} \Pi(a_i \mid u_{A_i}) = 1$ from Lemma 5.2)

Moreover, $U_{A_j} \subseteq \{A_{j+1}, ..., A_N\}$ implies : **(iii)** $\Pi_m(u_{A_j}) \geq \Pi_m(a_{j+1}, ..., a_N)$.
Then from (i), (ii) and (iii) we deduce that: $\Pi(a_j \mid u_{A_j}) > \min_{i=j+1..N} \Pi(a_i \mid u_{A_i})$.
Moreover, from the min-based chain rule, we have
$\pi_m(v) = min(\Pi(a_1 \mid u_{A_1}), ..., \Pi(a_j \mid u_{A_j}), ..., \Pi(a_N \mid u_{A_N}))$ thus,
$\pi_m(v) = \min_{i=1..N, i \neq j} \Pi(a_i \mid u_{A_i})$. This means that $\Pi(a_j \mid u_{A_j})$ does not intervene in the value of $\pi_m(v)$ and that we can substitute it by the degree 1.

Proof of Proposition 5.6 Let ΠG_m be a min-based possibilistic network. Let π_m be the joint possibility distribution computed using 5.6. We want to prove that each variable $A \in V$, is NI-independent of the variables in Z_A given its parent set U_A i.e. $\forall a \in D_A, \forall u_A \in D_{U_A}, \forall z_A \in D_{Z_A} : \Pi_m(a \wedge u_A \mid z_A) = min(\Pi_m(a \mid u_A), \Pi_m(u_A \mid z_A))$.
Let:
- a be a fixed instance of A, and $\neg a = D_A - \{a\}$
- u_A be a possible instantiation of the parent set U_A,
- x_A be a possible instantiation of the descendants set X_A,
- z_A be a possible instantiation of the non-descendants set Z_A,
- $\alpha_1 = \Pi(a \mid u_A)$,
- $\alpha_2 = \Pi(\neg a \mid u_A)$,
- $\beta_1 = min\{\Pi(b \mid u_B) : b \in u_A, u_B \subseteq z_A\}$,
- $\gamma_1 = min\{\Pi(c \mid u_C) : c \in z_A\}$,
- $\delta_1 = min\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq a \wedge x_A \wedge u_A \wedge z_A\}$,
- $\delta_2 = min\{\Pi(e \mid u_E) : e \in x_A, u_E \subseteq \neg a \wedge x_A \wedge u_A \wedge z_A\}$.

We first show that **(i)** $\Pi_m(a \wedge z_A \wedge u_A) = min(\Pi(a \mid u_A), \Pi_m(z_A \mid u_A))$
We have by definition : $\Pi_m(a \wedge z_A \wedge u_A) = max_{x_A} \pi_m(a \wedge u_A \wedge z_A \wedge x_A)$
$= max_{x_A} min(\alpha_1, \beta_1, \gamma_1, \delta_1)$ (From 5.6)
$=$ **(ii)** $min(\alpha_1, \beta_1, \gamma_1, max_{x_A} \delta_1)$
$= min[\alpha_1, \beta_1, \gamma_1]$ (from Lemma 5.2)
$= min[\alpha_1, Max\{min[\alpha_1, \beta_1, \gamma_1], min[\alpha_2, \beta_1, \gamma_1]\}]$
(since $max(\alpha_1, \alpha_2) = 1$ from the normalization condition)
$= min[\alpha_1, Max\{min[\alpha_1, \beta_1, \gamma_1, max_{x_A} \delta_1], min[\alpha_2, \beta_1, \gamma_1, max_{x_A} \delta_2]\}]$ (from Lemma 5.2)
$= min[\alpha_1, Max\{\Pi_m(a \wedge u_A \wedge z_A), \Pi_m(\neg a \wedge u_A, \wedge z_A)\}]$ (from (ii))

$= min[\alpha_1, \Pi_m(z_A \wedge u_A)]$ (by definition)
$= min[\Pi(a \mid u_A), \Pi_m(z_A \wedge u_A)]$ (by recovering the value of α_1)

Our aim is to prove that **(iii)** $\Pi_m(a \wedge z_A \mid u_A) = min(\Pi_m(a \mid u_A), \Pi_m(z_A \mid u_A))$.
Then, we can distinguish two cases:
- Case (a): $\Pi(a \wedge z_A \mid u_A) = 1$
$\Rightarrow \Pi_m(a \wedge z_A \wedge u_A) = \Pi_m(u_A)$ (from the conditioning definition)
$\Rightarrow \Pi_m(z_A \wedge u_A) = \Pi_m(u_A)$ and $\Pi_m(a \wedge u_A) = \Pi_m(u_A)$
(since $\Pi_m(a \wedge z_A \wedge u_A) \geq \Pi_m(z_A \wedge u_A)$ and $\Pi_m(a \wedge z_A \wedge u_A) \geq \Pi_m(a \wedge u_A)$)
$\Rightarrow \Pi_m(a \mid u_A) = \Pi_m(z_A \mid u_A) = 1$ (from the conditioning definition).
- Case (b): $\Pi(a \wedge z_A \mid u_A) \neq 1$
$\Rightarrow \Pi_m(a \wedge z_A \mid u_A) = \Pi_m(a \wedge z_A \wedge u_A) < \Pi_m(u_A)$ (from the conditioning definition)
Then, using (i) the expression (iii) is equivalent to :
(iv) $min(\Pi_m(a \mid u_A), \Pi_m(z_A \mid u_A)) = min(\Pi(a \mid u_A), \Pi_m(z_A \mid u_A))$.
To show (iv), we can distinguish two cases:

- $\Pi_m(z_A \mid u_A) \neq 1$
  $\Rightarrow \Pi_m(z_A \wedge u_A) = \Pi_m(z_A \mid u_A)$ (from the conditioning definition). Then,
  - if $\Pi_m(a \mid u_A) = \Pi(a \mid u_A)$ then the equality (iv) holds
  - if $\Pi_m(a \mid u_A) \neq \Pi(a \mid u_A)$
    $\Rightarrow \Pi_m(a \mid u_A) = 1$ and $\Pi(a \mid u_A) > \Pi_m(u_A) \geq \Pi_m(z_A \wedge u_A)$ (from Proposition 5.4)
    Hence the equality (iv) holds.
- $\Pi_m(z_A \mid u_A) = 1$
  $\Rightarrow \Pi_m(z_A \wedge u_A) = \Pi_m(u_A)$ (from the conditioning definition)
  In this case the expression (iv) is equivalent to :
  (v) $\Pi_m(a \mid u_A) = min(\Pi(a \mid u_A), \Pi_m(z_A \mid u_A))$
  Moreover, we have $min(\Pi(a \mid u_A)) = \Pi_m(a \wedge z_A \wedge u_A) < 1$ (from (i) and (b))
  Then $\Pi_m(a \mid u_A) < 1$
  $\Rightarrow \Pi_m(a \mid u_A) = \Pi(a \mid u_A) < \Pi_m(u_A)$ (from the conditioning definition)
  Hence the equality (v) holds.

Appendix D

Proofs of Chapter 6

Proofs relative to product-based possibilistic networks

Proof of Proposition 6.1 We want to prove that if a junction tree is globally consistent, then for each cluster C_i, $\pi_{C_i} = \Pi_p(C_i)$. This relation is true if the juntion tree contains a unique node. Suppose now that it is true for a junction tree with n nodes, thus we show that it is also true with (n+1) nodes.
Let C_i be a leaf of $\mathcal{JT}$ connected to the cluster C_j and let S_{ij} be their separator (see figure D.1). Let $\mathcal{JT}' = \mathcal{JT} \setminus C_i$ and V' be the universe relative to the junction tree $\mathcal{JT}'$.

Figure D.1: C_i is a leaf of the $\mathcal{JT}$ linked to C_j

We have (i) $\pi_{\mathcal{JT}} = \pi_{\mathcal{JT}'}.\frac{\pi_{C_i}}{\pi_{S_{ij}}}$ (from 6.6)

Let $D = C_i \setminus S_{ij}$ and $H = C_j \setminus S_{ij}$, then from the junction tree property: $D \cap V' = \emptyset$ (otherwise C_i is connected to an other cluster in $\mathcal{JT}'$ i.e $\mathcal{JT}' \cap C_i \neq \emptyset$ which contradicts $\mathcal{JT}' = \mathcal{JT} \setminus C_i$). Since the junction tree $\mathcal{JT}$ is consistent, then $max_D \pi_{C_i} = max_H \pi_{C_j} = \pi_{S_{ij}}$, thus:
$max_D \pi_{\mathcal{JT}}$
$= max_D(\pi_{\mathcal{JT}'} \cdot \frac{\pi_{C_i}}{\pi_{S_{ij}}})$ (from (i))
$= \pi_{\mathcal{JT}'}.\frac{max_D \pi_{C_i}}{\pi_{S_{ij}}}$ (since $D \cap V' = \emptyset$)
$= \pi_{\mathcal{JT}'}.\frac{\pi_{S_{ij}}}{\pi_{S_{ij}}}$ (from consistency property)
$= \pi_{\mathcal{JT}'}$

Thus $\pi_{\mathcal{JT}'}$ corresponds to the joint distribution relative to $\mathcal{JT}'$. Then from the induction

hypothesis we have, (ii) $max_{V \setminus C_k} \pi_{\mathcal{JT}} = \pi_{C_k}$, for all $C_K \in \mathcal{JT}'$.
Moreover, $max_{V \setminus C_i} \pi_{\mathcal{JT}}$
$= max_{V' \setminus S_{ij}} (\pi_{\mathcal{JT}'} . \frac{\pi_{C_i}}{\pi_{S_{ij}}})$ (since $V \setminus C_i = V' \setminus S_{ij}$)
$= max_{V' \setminus S_{ij}} \pi_{\mathcal{JT}'} . \frac{\pi_{C_i}}{\pi_{S_{ij}}}$ (since $C_i \cap (V' \setminus S_{ij}) = \emptyset$)

Moreover, $C_j \in \mathcal{JT}'$ then from (ii) we obtain, $max_{V \setminus C_j} \pi_{\mathcal{JT}} = \pi_{C_j}$, thus π_{C_j} corresponds to the local distribution on C_j. In addition, the marginalization of $\pi_{\mathcal{JT}'}$ on $V' \setminus S_{ij}$ is equivalent to the marginalization of π_{C_j} on S_{ij}, since S_{ij} only exists in C_j i.e.
$max_{V' \setminus S_{ij}} \pi_{\mathcal{JT}'} = max_{C_j \setminus S_{ij}} \pi_{C_j}$. This implies that:
$max_{V' \setminus S_{ij}} \pi_{\mathcal{JT}'} \cdot \frac{\pi_{C_i}}{\pi_{S_{ij}}} = max_{C_j \setminus S_{ij}} \pi_{C_j} . \frac{\pi_{C_i}}{\pi_{S_{ij}}}$
$= \pi_{S_{ij}} \cdot \frac{\pi_{C_i}}{\pi_{S_{ij}}}$ (from consistency property)
$= \pi_{C_i}$.

Proof of Proposition 6.2 Let ΠG_p be a min-based possibilistic network. Let $\mathcal{JT}$ be the junction tree corresponding to ΠG_p generated by the above initialization procedure. Let π_p be the joint distribution encoded by ΠG_p and $\pi^I_{\mathcal{JT}}$ be the joint distribution encoded by $\mathcal{JT}$ (using (6.6)). We want to prove that $\pi_p = \pi^I_{\mathcal{JT}}$. We have:

$$\pi^I_{\mathcal{JT}} = \frac{\prod_{i=1..m} \pi_{C_i}}{\prod_{j=1..m-1} \pi_{S_{ij}}}$$
$= \frac{\prod_{i=1..N} \Pi(A_i | U_{A_i})}{1}$ (from initialization procedure)
$= \pi_p$.

Proof of Proposition 6.3 Let $\pi^t_{\mathcal{JT}}$ be the joint distribution relative to a junction tree $\mathcal{JT}$ at level t. Let $\pi^{t+1}_{\mathcal{JT}}$ be the resulted joint distribution after the modification of a cluster C_i using the above procedure. We want to prove that $\pi^t_{\mathcal{JT}} = \pi^{t+1}_{\mathcal{JT}}$.

When a cluster C_i sends a message to a cluster C_j, then only the potentials of C_j and S_{ij} are changed. Therefore, to show this proof it is enough to prove that the fraction of C_j's and S_{ij}'s potentials remains unchanged i.e $\frac{\pi^{t+1}_{C_j}}{\pi^{t+1}_{S_{ij}}} = \frac{\pi^t_{C_j}}{\pi^t_{S_{ij}}}$.

Using (6.13) we have: $\frac{\pi^{t+1}_{C_j}}{\pi^{t+1}_{S_{ij}}} = \frac{\pi^t_{C_j} . \frac{\pi^{t+1}_{S_{ij}}}{\pi^t_{S_{ij}}}}{\pi^{t+1}_{S_{ij}}} = \frac{\pi^t_{C_j}}{\pi^t_{S_{ij}}}$.

Proof of Proposition 6.5 We want to prove that the collect and distribute phase are enough to make the junction tree globally consistent. This is equivalent to show that the link between

any two adjacent clusters C_i and C_j becomes consistent after the collect and distribute evidence phases.

We denote π'_{C_i} (resp. $\pi'_{S_{ij}}$) the potential of any cluster C_i (resp. separator S_{ij}) in the collect evidence phase and π''_{C_i} (resp. $\pi''_{S_{ij}}$) the potential of any cluster C_i (resp. separator S_{ij}) in the distribute evidence phase. Thus our aim is to show that for any cluster C_i, $\pi''_{C_i} = \pi^C_{C_i}$ and for any separator S_{ij}, $\pi''_{S_{ij}} = \pi^C_{S_{ij}}$. In other terms we should show that $\max_{C_i \setminus S_{ij}} \pi''_{C_i} = \pi''_{S_{ij}} = max_{C_j \setminus S_{ij}} \pi''_{C_j}$.

Let the first message to be passed between C_i and C_j (in collect phase) be from C_i to C_j then $\pi'_{S_{ij}} = \max_{C_i \setminus S_{ij}} \pi^I_{C_i}$. Next, when in the distribute evidence the message from C_j to C_i has to be passed, the potentials of S_{ij} and C_i have not been changed. Indeed, C_i has not received further messages since it sends message to C_j only if it has received messages from all its other adjacent clusters. Then,

$\max_{C_i \setminus S_{ij}} \pi''_{C_i} = \max_{C_i \setminus S_{ij}} (\pi'_{C_i} \cdot \frac{\pi''_{S_{ij}}}{\pi'_{S_{ij}}})$ (from (6.13))

$= \frac{\pi''_{S_{ij}}}{\pi'_{S_{ij}}} \cdot \max_{C_i \setminus S_{ij}} \pi'_{C_i}$

$= \frac{\pi''_{S_{ij}}}{\pi'_{S_{ij}}} \cdot \pi'_{S_{ij}}$ (since the potentials of S_{ij} and C_i have not been changed after C_i sends a message to C_j in the collect evidence phase i.e $\max_{C_i \setminus S_{ij}} \pi'_{C_i} = \pi'_{S_{ij}}$ from (6.11) and (6.12))

$= \pi''_{S_{ij}}$

$= \max_{C_j \setminus S_{ij}} \pi'_{C_j}$ (from (6.12))

$= \max_{C_j \setminus S_{ij}} \pi''_{C_j}$ (from (6.11))

Moreover, in the distribute evidence phase, each node receives only one message, therefore after receiving the message from C_j, the potential of C_i will not change. Therefore the link between C_i and C_j is consistent.

Proofs relative to min-based possibilistic networks

Proof of Proposition 6.6

1: We want to prove that in the context of any node A, the two sets E_A^+ and E_A^- are NI-independent.
Let $Y' = E_A^- \setminus E_A$. The variables in Y' are d-separated from E_A^+ by A. Indeed, $\forall B \in E_A^+, \forall C \in Y'$, the chain between B and C contains A and the arcs which determine that A is in this chain meet it head to tail (case 2 of d-separation)
$\Rightarrow$ (i) $\Pi(a \wedge e_A^+ \wedge y') = min(\Pi(a \wedge e_A^+), \Pi(a \wedge y')), \forall a$.
Moreover, (i) is equivalent to $\Pi(a \wedge e_A^+ \wedge y' \wedge e_A) = min(\Pi(a \wedge e_A^+), \Pi(a \wedge y' \wedge e_A)), \forall a$
$\Rightarrow \Pi(a \wedge e_A^+ \wedge e_A^-) = min(\Pi(a \wedge e_A^+), \Pi(a \wedge e_A^-)), \forall a$ (since $E_A^- = Y' \cup E_A$).

2: We want to prove that $\forall Y_i \in Y$, A d-separates $E_{AY_i}^-$ from $\{E_{AY_{i+1}}^-, ..., E_{AY_m}^-\}$ (see Figure 6.1). This relation corresponds to case 1 of d-separation. Indeed, $\forall B \in E_{AY_i}^-$,
$\forall C \in \{E_{AY_{i+1}}^-, ..., E_{AY_m}^-\}$ the chain between B and C contains A and the arcs which determine that A is in this chain meet it tail to tail.

3: We want to prove that E_A^+ and A are NI-independent in the context U. Let $U' = E_A^+/U$ be the instanciated variables in E_A^+ except the parent set of A and $U'' = U \cap E_A^+$ be the instanciated parents of A. The variables in U' are d-separated from A by the set U since $\forall B \in U'$, the chain between B and A contains a node $U_i \in U$ and the arcs which determine that U_i is in this chain meet it head to tail (case 2 of d-separation).
$\Rightarrow$ (i) $\Pi(u' \wedge a \wedge u) = min(\Pi(u' \wedge u), \Pi(a \wedge u)), \forall a, \forall u$

- If $U'' = \emptyset$ then (i) is equivalent to :
  $\Pi(e_A^+ \wedge a \wedge u) = min(\Pi(e_A^+ \wedge u), \Pi(a \wedge u)), \forall a, \forall u$ (since $U'' = \emptyset$ implies $e_A^+ = U'$)
- If $U'' \neq \emptyset$ then (i) is equivalent to :
  (ii) $\Pi(u' \wedge a \wedge u) = min(\Pi(u' \wedge u), \Pi(a \wedge u)), \forall a, \forall u$ s.t $u[U''] = e_A^+[U'']$
  Let $u'' = u[U'']$, then (ii) is equivalent to:
  $\Pi(u' \wedge a \wedge u \wedge u'') = min(\Pi(u' \wedge u \wedge u''), \Pi(a \wedge u)), \forall a, \forall u$ s.t $u[U''] = e_A^+[U'']$
  $\Rightarrow \Pi(e_A^+ \wedge a \wedge u) = min(\Pi(e_A^+ \wedge u), \Pi(a \wedge u)), \forall a, \forall u$ s.t $u[U''] = e_A^+[U'']$
  (since $E_A^+ = U' \cup U''$)

4: We want to prove that $\forall U_i \in U, U_i$ and $\{U_{i+1}, ..., U_n\}$ are NI-independent in the context E_A^+ (see Figure 6.2). Let U'=$\{U_{i+1}, ..., U_n\}$, $U'' = E_A^+ \cap U'$, $U''' = U' \setminus U''$ then:

- If U_i is not instanciated, then E_A^+ d-separates U_i and U''' (case 3 of d-separation indeed $\forall U_j \in U'''$ the chain between U_i and U_j contains A and the arcs which

determine that A is in this chain meet it head to head, and $A \notin E_A^+$ and $Y \notin E_A^+$)
$\Rightarrow \Pi(u_i \wedge u''' \wedge e_A^+) = min(\Pi(u_i \wedge e_A^+), \Pi(u''' \wedge e_A^+)), \forall u_i, \forall u'''$
$\Rightarrow \Pi(u_i \wedge u''' \wedge u'' \wedge e_A^+) = min(\Pi(u_i \wedge e_A^+), \Pi(u''' \wedge u'' \wedge e_A^+)), \forall u_i, \forall u'''$ where $u'' = e_A^+[U'']$
$\Rightarrow \Pi(u_i \wedge u' \wedge e_A^+) = min(\Pi(u_i \wedge e_A^+), \Pi(u' \wedge e_A^+)), \forall u_i, \forall u'$ where $u'[U''] = e_A^+[U'']$

- If U_i is instanciated (i.e $U_i \subseteq E_A^+$) we should verify that U_i and U' are independent in the context of E_A^+ i.e
  $\Pi(u_i \wedge u' \wedge e_A^+) = min(\Pi(u_i \wedge e_A^+), \Pi(u' \wedge e_A^+)), \forall u'$
  This is equivalent to prove that:
  $\Pi(u' \wedge e_A^+) = min(\Pi(e_A^+), \Pi(u', e_A^+), \forall u'$ (since U_i is instanciated i.e. $U_i \subseteq E_A^+$).
  This relation is true since $\Pi(e_A^+) \geq \Pi(u' \wedge e_A^+)$.

5: We want to prove that $\forall U_i \in U, U_i$ and $\{E^+_{U_{i+1}A}, ..., E^+_{U_nA}\}$ are NI-independent in the context $E^+_{U_iA}$ (see Figure 6.3). Let $U' = \{E^+_{U_{i+1}A}, ..., E^+_{U_nA}\}$, then:

- If U_i is not instanciated, then $E^+_{U_iA}$ d-separates U_i and U' (case 3 of d-separation indeed $\forall B \in U'$ the chain between U_i and B contains A and the arcs which determine that A is in this chain meet it head to head, and $A \notin E^+_{U_1A}$ and $Y \notin E^+_{U_1A}$).
- If U_i is instanciated (i.e $U_i \subseteq E^+_{U_iA}$) we should verify that U_i and U' are independent in the context of $E^+_{U_iA}$ i.e
  $\Pi(u_i \wedge u' \wedge e^+_{U_iA}) = min(\Pi(u_i \wedge e^+_{U_iA}), \Pi(u' \wedge e^+_{U_iA}))$
  This is equivalent to prove that:
  $\Pi(u', e^+_{U_iA}) = min(\Pi(e^+_{U_iA}), \Pi(u' \wedge e^+_{U_iA}))$ (since $U_i \subseteq E^+_{U_iA}$)
  This relation is true since $\Pi(e^+_{U_iA}) \geq \Pi(u' \wedge e^+_{U_iA})$.

6: We want to prove that $\forall U_i \in U$, $E^+_{U_iA}$ is d-separated from $\{E^+_{U_{i+1}A}, ..., E^+_{U_nA}\}$.
$\forall B \in E^+_{U_iA}, \forall C \in \{E^+_{U_{i+1}A}, ..., E^+_{U_nA}\}$ the chain between B and C contains A and the arcs which determine that A is in this chain meet it head to head, and $A \notin \emptyset$ and $Y \not\subset \emptyset$.

7: We want to prove that given a node A with two parents U_i and V (see Figure 6.4), then the two sets E^+_{VA} and E^-_A are NI-independent in the context of the nodes $\{A, U_i, V\}$.
Let V'$= E^+_{VA} \setminus V$ and $Y' = E^-_A \setminus E_A$. The variables in V' are d-separated from Y' by the set $\{A, U_i, V\}$ since $\forall B \in V'$, $\forall C \in Y'$, the chain between B and C contains A and the arcs which determine that A is in this chain meet it head to tail (case 2 of d-separation).
$\Rightarrow$ (i) $\Pi(v' \wedge y' \wedge a \wedge u_i \wedge v) = min(\Pi(v' \wedge a \wedge u_i \wedge v), \Pi(y' \wedge a \wedge u_i \wedge v)), \forall a, \forall u_i, \forall v$

- If V is not instanciated, then (i) is equivalent to:
  $\Pi(v' \wedge v \wedge y' \wedge a \wedge u_i \wedge e_A) = min(\Pi(v' \wedge v \wedge a \wedge u_i), \Pi(y' \wedge a \wedge u_i \wedge e_A)), \forall a, \forall u_i, \forall v$

$\Rightarrow \Pi(e^+_{VA} \wedge e^-_A \wedge a \wedge u_i \wedge v) = min(\Pi(e^+_{VA} \wedge a \wedge u_i \wedge v), \Pi(e^-_A \wedge a \wedge u_i \wedge v)), \forall a, \forall u_i, \forall v$
(since V is not instanciated implies that $E^+_{VA} = V'$ and $E^-_A = Y' \cup E_A$)

- If V is instanciated, then (i) is equivalent to:
  $\Pi(v' \wedge y' \wedge a \wedge u_i \wedge v) = min(\Pi(v' \wedge a \wedge u_i \wedge v), \Pi(y' \wedge a \wedge u_i \wedge v)), \forall a, \forall u_i$ where $v = e^+_{VA}[V]$
  $\Rightarrow \Pi(v' \wedge y' \wedge a \wedge u_i \wedge v \wedge v) = min(\Pi(v' \wedge a \wedge u_i \wedge v \wedge v), \Pi(y' \wedge a \wedge u_i \wedge v)), \forall a, \forall u_i$ where $v = e^+_{VA}[V]$ (since $\Pi(v \wedge v) = \Pi(v)$)
  $\Rightarrow \Pi(e^+_{VA} \wedge e^-_A \wedge a \wedge u_i \wedge v) = min(\Pi(e^+_{VA} \wedge a \wedge u_i \wedge v), \Pi(e^-_A \wedge a \wedge u_i \wedge v)), \forall a, \forall u_i, \forall v$
  (since $E^+_{VA} = V' \cup V$ and $E^-_A = Y' \cup E_A$)

8: We want to prove that given a node A with two parents U_i and V (see Figure 6.5), then the two sets E^-_A and $U = \{U_i, V\}$ are NI-independent in the context of A.
Let $Y' = E^-_A \setminus E_A$. The variables in Y' are d-separated from $U = \{U_i, V\}$ by A. Indeed, $\forall B \in Y', \forall C \in U$, the chain between B and C contains A and the arcs which determine that A is in this chain meet it head to tail (case 2 of d-separation)
$\Rightarrow$ (i) $\Pi(a \wedge u, y') = min(\Pi(a \wedge u), \Pi(a \wedge y')), \forall a, \forall u$.
Then (i) is equivalent to $\Pi(a \wedge u \wedge y' \wedge e_A) = min(\Pi(a \wedge u), \Pi(a \wedge y' \wedge e_A)), \forall a, \forall u$
$\Rightarrow \Pi(a \wedge u \wedge e^-_A) = min(\Pi(a \wedge u), \Pi(a \wedge e^-_A)), \forall a, \forall u$ since $E^-_A = Y' \cup E_A$.

9: We want to prove that given a node A with two parents U_i and V, then the two sets E^+_{VA} and $\{U_i, A\}$ are NI-independent in the context of V.
Let V'$= E^+_{VA}/V$. The variables in V' are d-separated from $\{U_i, A\}$ by V since $\forall B \in V'$, $\forall C \in \{U_i, A\}$, the chain between B and C contains V and the arcs which determine that V is in this chain meet it head to tail (case 2 of d-separation).
$\Rightarrow$ (i) $\Pi(v' \wedge u_i \wedge a \wedge v) = min(\Pi(v' \wedge v), \Pi(u_i \wedge a \wedge v)), \forall u_i, \forall a, \forall v$

- If V is not instanciated, then (i) is equivalent to:
  $\Pi(e^+_{VA} \wedge u_i \wedge a \wedge v) = min(\Pi(e^+_{VA} \wedge v), \Pi(u_i \wedge a \wedge v)), \forall u_i, \forall a, \forall v$
  (since V is not instanciated implies that $E^+_{VA} = V'$)
- If V is instanciated, then (i) is equivalent to:
  $\Pi(v' \wedge v \wedge u_i \wedge a \wedge v) = min(\Pi(v' \wedge v \wedge v), \Pi(u_i \wedge a \wedge v)), \forall u_i, \forall a$ where $v = e^+_{VA}[V]$
  (since $\Pi(v \wedge v) = \Pi(v)$)
  $\Rightarrow \Pi(e^+_{VA} \wedge u_i \wedge a \wedge v) = min(\Pi(e^+_{VA} \wedge v), \Pi(u_i \wedge a \wedge v)), \forall u_i, \forall a$ where $v = e^+_{VA}[V]$
  (since $E^+_{VA} = V' \cup V$)

Proof of Lemma 6.1 $\forall a \in D_A$, the current joint possibility measure of a based on the total evidence e is defined by $Bel_{Joint}(a) = \Pi_m(a \wedge e) = \Pi_m(a \wedge e^-_A, e^+_A)$

$= min(\Pi_m(a \wedge e_A^-), \Pi_m(a \wedge e_A^+))$
since E_A^+ and E_A^- are NI-independent in the context of A (From relation 1 in Proposition 6.6).
Let $\lambda(a) = \Pi_m(a \wedge e_A^-)$ and $\mu(a) = \Pi_m(a \wedge e_A^+)$, then $Bel_{Joint}(a) = min(\lambda(a), \mu(a))$.

Proof of Lemma 6.2 The λ value $\forall a \in D_A$ is defined by $\lambda(a) = \Pi_m(a \wedge e_A^-) = \Pi_m(a \wedge e_A \wedge e_{AY_1}^- \wedge ... \wedge e_{AY_m}^-)$.
Thus, if A is not instanciated then:
$\lambda(a) = \Pi_m(a \wedge e_{AY_1}^- \wedge ... \wedge e_{AY_m}^-) = min(\Pi_m(a), \Pi_m(a \wedge e_{AY_1}^-), \Pi_m(a \wedge e_{AY_2}^- \wedge ... \wedge e_{AY_m}^-))$
Since $\forall Y_i \in Y$, A d-separates $E_{AY_i}^-$ from $\{E_{AY_{i+1}}^-, ..., E_{AY_m}^-\}$ (From relation 2 in Proposition 6.6). When iterating the same operation on $\Pi_m(a \wedge e_{AY_2}^- \wedge ... \wedge e_{AY_m}^-)$, we obtain:
$\lambda(a) = min(\Pi_m(a \wedge e_{AY_1}^-), \Pi_m(a \wedge e_{AY_2}^-), ..., \Pi_m(a \wedge e_{AY_m}^-)$
Let $\lambda_{Y_j}(a) = \Pi_m(a \wedge e_{AY_j}^-)$ then $\lambda(a) = min(\min_{j=1}^m \lambda_{Y_j}(a))$.

Instead of creating dummy or evidence nodes we consider that local evidence is stored within each node. Let $\lambda_A(a)$ be the local evidence related to the node A such that:

$$\lambda_A(a) = \begin{cases} 0 & \text{if } e_A \neq a \text{ (A is instanciated to } (e_A \neq a) \\ 1 & \text{otherwise (A is instanciated to a } (e_A = a) \text{ or A is not instanciated)} \end{cases}$$

Thus a more general formula of $\lambda(a)$ is defined by: $\lambda(a) = min(\lambda_A(a), \min_{j=1}^m \lambda_{Y_j}(a))$.

Proof of Lemma 6.3 The μ value $\forall a \in D_A$ is defined by:
$\mu(a) = \Pi_m(a \wedge e_A^+)$
$= \max_u \Pi_m(a \wedge e_A^+ \wedge u)$
$= \max_u(min(\Pi_m(e_A^+ \wedge u), \Pi_m(a \wedge u)))$
since E_A^+ and A are NI-independent in the context U (From relation 3 in Proposition 6.6).
Moreover, $\forall U_i \in U, U_i$ and $\{U_{i+1}, ..., U_n\}$ are NI-independent in the context E_A^+ (From relation 4 in Proposition 6.6), then:
$\Pi_m(e_A^+ \wedge u) = \Pi_m(u_1 \wedge ... \wedge u_n \wedge e_A^+)$
$= min(\Pi_m(u_1 \wedge e_A^+), \Pi_m(u_2 \wedge ... \wedge u_n \wedge e_A^+))$
$= min(\Pi_m(u_1 \wedge e_{U_1A}^+ \wedge ... \wedge e_{U_nA}^+), \Pi_m(u_2 \wedge ... \wedge u_n \wedge e_{U_1A}^+ \wedge ... \wedge e_{U_nA}^+))$
$= min(min(\Pi_m(u_1 \wedge e_{U_1A}^+), \Pi_m(e_{U_1A}^+ \wedge ... \wedge e_{U_nA}^+)), \Pi_m(u_2 \wedge ... \wedge u_n \wedge e_{U_1A}^+ \wedge ... \wedge e_{U_nA}^+))$

Since $\forall U_i \in U, U_i$ and $\{E_{U_{i+1}A}^+, ..., E_{U_nA}^+\}$ are NI-independent in the context $E_{U_iA}^+$ (From relation 5 in Proposition 6.6), then:
$\Pi_m(e_A^+ \wedge u) = min(\Pi_m(u_1 \wedge e_{U_1A}^+), \Pi_m(e_{U_1A}^+ \wedge ... \wedge e_{U_nA}^+), \Pi_m(u_2 \wedge ... \wedge u_n \wedge e_{U_1A}^+ \wedge ... \wedge e_{U_nA}^+))$
$= min(\Pi_m(u_1 \wedge e_{U_1A}^+), \Pi_m(u_2 \wedge ... \wedge u_n \wedge e_{U_1A}^+ \wedge ... \wedge e_{U_nA}^+))$

(since $\Pi_m(e^+_{U_1A} \wedge ... \wedge e^+_{U_nA}) \geq \Pi_m(u_2 \wedge ... \wedge u_n \wedge e^+_{U_1A} \wedge ... \wedge e^+_{U_nA})$).

If we iterate the same operation on $\Pi_m(u_2 \wedge ... \wedge u_n, e^+_{U_1A} \wedge ... \wedge e^+_{U_nA}))$, we obtain:
$\Pi_m(e^+_A \wedge u) = min(\Pi_m(u_1 \wedge e^+_{U_1A}), \Pi_m(u_2 \wedge e^+_{U_2A})...\Pi_m(u_n \wedge e^+_{U_nA}), \Pi_m(e^+_{U_1A} \wedge ... \wedge e^+_{U_nA}))$

Moreover, $\Pi_m(e^+_{U_1A} \wedge ... \wedge e^+_{U_nA}) = min(\Pi_m(e^+_{U_1A}), \Pi_m(e^+_{U_2A} \wedge ... \wedge e^+_{U_nA}))$
since $\forall U_i \in U$, $E^+_{U_iA}$ is d-separated from $\{E^+_{U_{i+1}A}, ..., E^+_{U_nA}\}$ (From relation 6 in Proposition 6.6). Then, if we iterate the same operation on $\Pi_m(e^+_{U_2A} \wedge ... \wedge e^+_{U_nA})$, we obtain:
$\Pi_m(e^+_{U_1A} \wedge ... \wedge e^+_{U_nA}) = min(\Pi_m(e^+_{U_1A}), ..., \Pi_m(e^+_{U_nA}))$. Then,
$\Pi_m(e^+_A \wedge u)$
$= min(\Pi_m(u_1 \wedge e^+_{U_1A}), \Pi_m(u_2 \wedge e^+_{U_2A}), ..., \Pi_m(u_n \wedge e^+_{U_nA}), min(\Pi_m(e^+_{U_1A}), ..., \Pi_m(e^+_{U_nA})))$
$= min(\Pi_m(u_1 \wedge e^+_{U_1A}), \Pi_m(u_2 \wedge e^+_{U_2A}), ..., \Pi_m(u_n \wedge e^+_{U_nA}), \Pi_m(e^+_{U_1A}), ..., \Pi_m(e^+_{U_nA}))$
$= min(\Pi_m(u_1 \wedge e^+_{U_1A}), \Pi_m(u_2 \wedge e^+_{U_2A}), ..., \Pi_m(u_n \wedge e^+_{U_nA}))$
(since $\forall i \in \{1..n\}, \Pi_m(e^+_{U_iA}) \geq \Pi_m(u_i \wedge e^+_{U_iA})$)
Then, $\mu(a) = \max_u min(\Pi_m(a \wedge u), \min_{i=1}^n \Pi_m(u_i \wedge e^+_{U_iA}))$
$= \max_u min(\Pi(a \wedge u), \min_{i=1}^n \Pi_m(u_i \wedge e^+_{U_iA}))$.

We substitute here $\Pi_m(a \wedge u)$ by $\Pi(a \wedge u)$ since we show later that if the initial distributions ($\Pi(a \wedge u)$) are not coherent with the axioms of the possibility theory then they are eliminated by the minimum operator. Moreover we have shown in Chapter 5 that in the incoherence case $\Pi_m(a \wedge u) = 1$ which means that this value is also eliminated by the minimum operator.
Let $\mu_A(u_i) = \Pi_m(u_i \wedge e^+_{U_iA})$, then $\mu(a) = \max_u min(\Pi(a \wedge u), \min_{i=1}^n \mu_A(u_i))$.

Proof of Lemma 6.4 In order to compute $\lambda_A(u_i)$, we will proceed in the same manner than the probabilistic case by considering that all parents of A except U_i form a single node $V = U - U_i$ (see Figure D.2). Then, the set $E^-_{U_iA}$ can be decomposed into e^-_A (depending on A's children) and E^+_{VA} (depending on A's parents except U_i): $E^-_{U_iA} = E^+_{VA} \cup E^-_A$ where $E^+_{VA} = \bigcup_{k \neq i} E^+_{U_kA}$.

Figure D.2: Evidence used in the derivation of $\lambda_A(U_i)$

The λ message from A to its parent $U_i, (i \in \{1, ..n\})$ when $U_i = u_i$ is defined by $\lambda_A(u_i) = \Pi_m(e^-_{U_iA} \wedge u_i)$. Thus when considering that $E^-_{U_iA} = E^+_{VA} \cup E^-_A$, we have:
$\lambda_A(u_i) = \Pi_m(e^+_{VA} \wedge e^-_A \wedge u_i) = \max_a \max_v \Pi_m(e^+_{VA} \wedge e^-_A \wedge u_i \wedge a \wedge v)$

The two sets E^+_{VA} and E^-_A are NI-independent in the context of the nodes $\{A, U_i, V\}$ (From relation 7 in Proposition 6.6), then:
$\Pi_m(e^+_{VA} \wedge e^-_A \wedge a \wedge u_i \wedge v) = min(\Pi_m(e^+_{VA} \wedge a \wedge u_i \wedge v), \Pi_m(e^-_A \wedge a \wedge u_i \wedge v))$
$= min(\Pi_m(e^+_{VA} \wedge u_i \wedge a \wedge v), min(\Pi_m(e^-_A \wedge a), \Pi_m(u_i \wedge a, v)))$
since E^-_A and $U = \{U_i, V\}$ are NI-independent in the context of A (From relation 8 in Proposition 6.6).
Moreover E^+_{VA} and $\{U_i, A\}$ are NI-independent in the context of V (From relation 9 in Proposition 6.6), then:
$\Pi_m(e^+_{VA} \wedge e^-_A \wedge u_i \wedge a \wedge v)$
$= min[min(\Pi_m(e^+_{VA} \wedge v), \Pi_m(u_i \wedge a \wedge v)), min(\Pi_m(e^-_A \wedge a), \Pi_m(u_i \wedge a \wedge v))]$
$= min[\Pi_m(e^+_{VA} \wedge v), \Pi_m(u_i \wedge a \wedge v), \Pi_m(e^-_A \wedge a), \Pi_m(u_i \wedge a \wedge v)]$
$= min[\Pi_m(e^+_{VA} \wedge v), \Pi_m(e^-_A \wedge a), \Pi_m(u_i \wedge a \wedge v)]$, then:
$\lambda_A(u_i) = \max_a \max_v min(\Pi_m(e^+_{VA} \wedge v), \Pi_m(e^-_A \wedge a), \Pi_m(u_i \wedge a \wedge v))$
When restoring the meaning of V, we obtain:
$\Pi_m(u_i \wedge a \wedge v) = \Pi_m(a \wedge u)$
$\Pi_m(e^+_{VA} \wedge v) = \min_{k \neq i} \Pi_m(u_k \wedge e^+_{U_k A}) = \min_{k \neq i} \mu_A(u_k)$

Moreover $\Pi_m(e^-_A \wedge a) = \lambda(a)$, then:
$\max_a \max_v min[\Pi_m(e^+_{VA} \wedge v), \Pi_m(e^-_A \wedge a), \Pi_m(u_i \wedge a \wedge v)]$
$= \max_a \max_{u_k : k \neq i} min[\lambda(a), \min_{k \neq i} \mu_A(u_k), \Pi_m(a \wedge u)]$
$= \max_a \max_{u_k : k \neq i} min[\lambda(a), min(\min_{k \neq i} \mu_A(u_k), \Pi_m(a \wedge u))]$
$= \max_a min[\lambda(a), \max_{u_k : k \neq i}(min(\min_{k \neq i} \mu_A(u_k), \Pi_m(a \wedge u)))]$
(since λ_A does not depend on u_k)
$= \max_a min[\lambda(a), \max_{u_k : k \neq i}(min(\Pi_m(a \wedge u), \min_{k \neq i} \mu_A(u_k)))]$
$= \max_a min[\lambda(a), \max_{u_k : k \neq i}(min(\Pi(a \wedge u), \min_{k \neq i} \mu_A(u_k)))]$.

We substitute here $\Pi_m(a \wedge u)$ by $\Pi(a \wedge u)$ since we show later that if the initial distributions $(\Pi(a \wedge u))$ are not coherent with the axioms of the possibility theory then they are eliminated by the minimum operator. Moreover we have shown in Chapter 5 that in the incoherence case $\Pi_m(a \wedge u) = 1$ which means that this value is also eliminated by the minimum operator.

Proof of Lemma 6.5 The μ message from A to its child $Y_j, (j \in \{1, .., m\})$ when $A = a$ is expressed by $\mu_{Y_j}(a) = \Pi_m(a \wedge e^+_{AY_j})$. This value can be computed by splitting the entire evidence e into $E^+_{AY_j}$ and $E^-_{AY_j}$. For instance, in Example 4.4, if we consider the variable A and its child H then, $E = E_{AH^+} \cup E_{AH^-} = \{A, D, F, J, I, M\}$ where $E_{AH^+} = \{A, D, F, J\}$

and $E_{AH^-} = \{I, M\}$.

Therefore, $e^+_{AY_j} = e/e^-_{AY_j}$ so $\mu_{Y_j}(a) = \Pi_m(a \wedge e/e^-_{AY_j})$.
Moreover, $Bel_{Joint}(a) = \Pi_m(a \wedge e) = min(\lambda(a), \mu(a))$
$= min(min(\lambda_A(a), \min_{i=1}^m \lambda_{Y_i}(a)), \mu(a))$
$= min(\lambda_A(a), \min_{i=1}^m \lambda_{Y_i}(a), \mu(a))$ (from (6.20) and (6.22)), then
$\mu_{Y_j}(a) = \Pi_m(a \wedge e/e^-_{AY_j}) = min(\lambda_A(a), \min_{i=1..m, i\neq j} \lambda_{Y_i}(a), \mu(a))$.

Proof of Proposition 6.7 We want to prove that if a junction tree is globally consistent, then for each cluster C_i, $\pi_{C_i} = \Pi_m(C_i)$. This relation is true if the juntion tree contains a unique node. Suppose now that it is true for a junction tree with n nodes thus we show that it is also true with (n+1) nodes.
Let C_i be a leaf of $\mathcal{JT}$ connected to the cluster C_j and let S_{ij} be their separator (see figure D.1). Let $\mathcal{JT}' = \mathcal{JT} \setminus C_i$ and V' be the universe relative to the junction tree $\mathcal{JT}'$. We have
(i) $\pi_{\mathcal{JT}} = min(\pi_{\mathcal{JT}'}, \pi_{C_i})$ (from 6.26)

Let $D = C_i \setminus S_{ij}$ and $H = C_j \setminus S_{ij}$, then from the junction tree property: $D \cap V' = \emptyset$ (otherwise C_i is connected to an other cluster in $\mathcal{JT}'$ i.e $\mathcal{JT}' \cap C_i \neq \emptyset$ which contradicts $\mathcal{JT}' = \mathcal{JT} \setminus C_i$).

Since the junction tree $\mathcal{JT}$ is consistent, then $max_D \pi_{C_i} = max_H \pi_{C_j} = \pi_{S_{ij}}$, thus:
$max_D \pi_{\mathcal{JT}} = max_D min(\pi_{\mathcal{JT}'}, \pi_{C_i})$ (from (i))
$= min(\pi_{\mathcal{JT}'}, max_D \pi_{C_i})$ (since $D \cap V' = \emptyset$)
$= min(\pi_{\mathcal{JT}'}, \pi_{S_{ij}})$ (from consistency property)
$= \pi_{\mathcal{JT}'}$ (Indeed, suppose that $\pi_{S_{ij}} < \pi_{\mathcal{JT}'}$ for a particular instance s_{ij}, then $\pi_{S_{ij}} < \pi_{C_j}$ which contradicts the consistency property.

Thus $\pi_{\mathcal{JT}'}$ corresponds to the joint distribution relative to $\mathcal{JT}'$. Then from the induction hypothesis, (ii) $max_{V \setminus C_k} \pi_{\mathcal{JT}} = \pi_{C_k}$, for all $C_K \in \mathcal{JT}'$.
Moreover, $max_{V \setminus C_i} \pi_{\mathcal{JT}} = max_{V' \setminus S_{ij}} min(\pi_{\mathcal{JT}'}, \pi_{C_i})$ (since $V \setminus C_i = V' \setminus S_{ij}$)
$= min(\pi_{C_i}, max_{V' \setminus S_{ij}} \pi_{\mathcal{JT}'})$ (since $C_i \cap (V' \setminus S_{ij}) = \emptyset$)

Moreover, $C_j \in \mathcal{JT}'$ then from (ii) we obtain, $max_{V \setminus C_j} \pi_{\mathcal{JT}} = \pi_{C_j}$, thus π_{C_j} corresponds to the local distribution on C_j. In addition, the marginalization of $\pi_{\mathcal{JT}'}$ on $V' \setminus S_{ij}$ is equivalent to the marginalization of π_{C_j} on S_{ij}, since S_{ij} only exists in C_j i.e.
$max_{V' \setminus S_{ij}} \pi_{\mathcal{JT}'} = max_{C_j \setminus S_{ij}} \pi_{C_j}$

This implies that, $min(\pi_{C_i}, max_{V' \setminus S_{ij}} \pi_{\mathcal{JT}'}) = min(\pi_{C_i}, max_{C_j \setminus S_{ij}} \pi_{C_j})$
$= min(\pi_{C_i}, \pi_{S_{ij}})$ (from consistency property)
$= \pi_{C_i}$ (since $\pi_{S_{ij}} = max_D \pi_{C_i}$).

Proof of Proposition 6.8 Let ΠG_m be a min-based possibilistic network. Let $\mathcal{JT}$ be the junction tree corresponding to ΠG_m using the above initialization procedure. Let π_m be the joint distribution encoded by ΠG_m and $\pi^I_{\mathcal{JT}}$ be the joint distribution encoded by $\mathcal{JT}$ (using (6.26)). We want to prove that $\pi_m = \pi^I_{\mathcal{JT}}$. We have: $\pi^I_{\mathcal{JT}} = min_{i=1..m} \pi^I_{C_i}$
$= min(min_{i=1..m} \Pi(A_i \mid U_{A_i}), 1)$ (from initialization procedure)
$= min_{i=1..m} \Pi(A_i \mid U_{A_i})$
$= \pi_m$.

Proof of Proposition 6.9 Let $\pi^t_{\mathcal{JT}}$ be the joint distribution relative to a junction tree $\mathcal{JT}$ at level t. Let $\pi^{t+1}_{\mathcal{JT}}$ be the resulted joint distribution after the modification of a cluster C_i using the above procedure. We want to prove that $\pi^t_{\mathcal{JT}} = \pi^{t+1}_{\mathcal{JT}}$.
To show this proof it is enough to prove that when a cluster C_i sends a message to a cluster C_j, then $\min_{k=1..m} \pi^t_{C_k} = min(\min_{k=1..m} \pi^t_{C_k}, \pi^{t+1}_{C_j})$ since only the potential of C_j is changed in the expression of the joint distribution.
$min(\min_{k=1..m} \pi^t_{C_k}, \pi^{t+1}_{C_j}) = min(\min_{k=1..m} \pi^t_{C_k}, min(\pi^t_{C_j}, \pi^{t+1}_{S_{ij}}))$ (from 6.32)
$= min(\min_{k=1..m} \pi^t_{C_k}, \pi^t_{C_j}, \pi^{t+1}_{S_{ij}})$
$= min(\min_{k=1..m} \pi^t_{C_k}, \pi^{t+1}_{S_{ij}})$ (since $\pi^t_{C_j} \leq \min_{k=1..m} \pi^t_{C_k}$. Indeed, $\pi^t_{C_j}$ is included in the expression $\min_{k=1..m} \pi^t_{C_k}$)
$= min(\min_{k=1..m} \pi^t_{C_k}, \max_{C_i \setminus S_{ij}} \pi^t_{C_i})$ (from 6.31)
$= \min_{k=1..m} \pi^t_{C_k}$ (since $\max_{C_i \setminus S_{ij}} \pi^t_{C_i} \leq \pi^t_{C_i} \leq \min_{k=1..m} \pi^t_{C_k}$).

Proof of Proposition 6.11 We want to prove that the collect and distribute phase are enough to make the junction tree globally consistent. This is equivalent to show that the link between any two adjacent clusters C_i and C_j becomes consistent after the collect and distribute evidence phases.
We denote π'_{C_i} (resp. $\pi'_{S_{ij}}$) the potential of any cluster C_i (resp. separator S_{ij}) in the collect evidence phase and π''_{C_i} (resp. $\pi''_{S_{ij}}$) the potential of any cluster C_i (resp. separator S_{ij}) in the distribute evidence phase. Thus our aim is to show that for any cluster C_i, $\pi''_{C_i} = \pi^C_{C_i}$ and for any separator S_{ij}, $\pi''_{S_{ij}} = \pi^C_{S_{ij}}$. In other terms we should show that $\max_{C_i \setminus S_{ij}} \pi''_{C_i} = \pi''_{S_{ij}} = max_{C_j \setminus S_{ij}} \pi''_{C_j}$.

Let the first message to be passed between C_i and C_j (in collect phase) be from C_i to C_j then $\pi'_{S_{ij}} = \max_{C_i \setminus S_{ij}} \pi^I_{C_i}$. Next, when in the distribute evidence the message from C_j to C_i has to be passed, the potentials of S_{ij} and C_i have not been changed. Indeed, C_i has not received further messages since it sends message to C_j only if it has received messages from all its other adjacent clusters. Then,
$\max_{C_i \setminus S_{ij}} \pi''_{C_i} = \max_{C_i \setminus S_{ij}} min(\pi'_{C_i}, \pi''_{S_{ij}})$ (from (6.32))
$= min(\pi''_{S_{ij}}, \max_{C_i \setminus S_{ij}} \pi'_{C_i})$
$= min(\pi''_{S_{ij}}, \pi'_{S_{ij}})$ (since the potentials of S_{ij} and C_i have not been changed after C_i sends a message to C_j in the collect evidence phase i.e $\max_{C_i \setminus S_{ij}} \pi'_{C_i} = \pi'_{S_{ij}}$ from (6.30) and (6.31))
$= \pi''_{S_{ij}}$
$= \max_{C_j \setminus S_{ij}} \pi'_{C_j}$ (from (6.31))
$= \max_{C_j \setminus S_{ij}} \pi''_{C_j}$ (from (6.30))

Moreover, in the distribute evidence phase, each node receives only one message, therefore after receiving the message from C_j, the potential of C_i will not change. Therefore the link between C_i and C_j is consistent.

Appendix E

Proofs of Chapter 7

Proof of Proposition 7.1 Let $\alpha\Pi G_m$ be an α-normalized min-based possibilistic network. Let π_m be the joint distribution computed from (5.6). We want to prove that $h(\pi_m) = \alpha$. To show this proof, it is enough to find a particular instance v of V, such that $\Pi(a \mid u_A) = \alpha, \forall a \in v$ and $u_A \subseteq v$. Such v can be obtained, in a constructive way, as follows:

Algorithm E.1: Construction of the instance v

begin
  Let v be a global instantiation of V containing initially s.t. $v \leftarrow \emptyset$;
  while $V \neq \emptyset$ **do**
    - Select a variable A of V such that A has no parent in V (First, we start by roots).;
    - Select an instance a of A such that $\Pi(a \mid u_A) = \alpha$ and $u_A \subseteq v$ (such instance always exists due to the normalization constraint of α-normalized min-based possibilistic networks);
    - $v \leftarrow v \wedge a$;
    - $V \leftarrow V - \{A\}$;
end

At the end of this procedure, all the variables in V are instanciated and we have $\forall a \in v$: $\Pi(a \mid u_A) = \alpha$ and $u_A \subseteq v$. Therefore, $min\{\Pi(a \mid u_A) : a \in v \text{ and } u_A \in v) = \alpha$, hence $h(\pi_m) = \alpha$.

Proof of Proposition 7.2 Let $\alpha\Pi G_m$ be an α-normalized min-based possibilistic network. Let $\mathcal{MG}$ be the moral graph corresponding to $\alpha\Pi G_m$ given by the initialization procedure. Let π_a be the joint distribution given by (7.2) (which is obtained after incorporating the instance a of the variable of interest A). Let $\pi^I_{\mathcal{MG}}$ be the joint distribution encoded by $\mathcal{MG}$ (given by (7.4)). We want to prove that $\pi_a = \pi^I_{\mathcal{MG}}$.

- Let $\pi^I_{\mathcal{MG}}$ be the joint distribution encoded by $\mathcal{MG}$ after the transformation of the initial

DAG into a moral graph (step 1 of the initialization procedure), then from (7.4) :
$\pi^I_{\mathcal{MG}} = \min_{i=1..N} \pi^I_{C_i}$
$= min_{i=1..N} \Pi(A_i \mid U_i)$ (since from the initialization procedure : $\forall C_i, \pi^I_{C_i} = \Pi(A_i \mid U_i)$)
$= \pi_m$ (From (5.6))

- Let $\pi^I_{C_i}$ be the potential of the cluster C_i relative to the variable of interest A_i after its incorporation (step 2 of the initialization procedure). Let $\pi^I_{\mathcal{MG}}$ be the joint distribution encoded by $\mathcal{MG}$ after updating C_i using (7.4), then
$\pi^I_{\mathcal{MG}} = min(\pi^I_{\mathcal{MG}}, \pi^I_{C_i})$. Thus from (7.4) we have:

$$\pi^I_{\mathcal{MG}}(\omega) = \begin{cases} \pi^I_{\mathcal{MG}}(\omega) & \text{if } \omega[A_i] = a_i \\ 0 & \text{otherwise} \end{cases}$$
$$= \begin{cases} \pi_m(\omega) & \text{if } \omega[A_i] = a_i \\ 0 & \text{otherwise} \end{cases} \quad \text{(since } \pi^I_{\mathcal{MG}} = \pi_m \text{ as shown before)}$$
$= \pi_a(\omega)$ (using (7.2))
$\Rightarrow \pi^I_{\mathcal{MG}} = \pi_a$.

Proof of Proposition 7.3 Let $\pi^t_{\mathcal{MG}}$ be the joint distribution relative to a moral graph $\mathcal{MG}$ at level t. Let $\pi^{t+1}_{\mathcal{MG}}$ be the resulted joint distribution after the modification of two parents C_i and C_j using equations (7.8), (7.9) and (7.10). We want to prove that $\pi^t_{\mathcal{MG}} = \pi^{t+1}_{\mathcal{MG}}$.
To show this proof it is enough to prove that $min(\pi^{t+1}_{C_i}, \pi^{t+1}_{C_j}) = min(\pi^t_{C_i}, \pi^t_{C_j})$. This relation is true, indeed:
$min(\pi^{t+1}_{C_i}, \pi^{t+1}_{C_j})$
$= min(min(\pi^{t+1}_{S_{ij}}, \pi^t_{C_i}), min(\pi^{t+1}_{S_{ij}}, \pi^t_{C_j}))$ (From (7.9) and (7.10))
$= min(\pi^t_{C_i}, \pi^t_{C_j}, \pi^{t+1}_{S_{ij}})$
$= min(\pi^t_{C_i}, \pi^t_{C_j}, min(max_{C_i \setminus S_{ij}} \pi^t_{C_i}, max_{C_j \setminus S_{ij}} \pi^t_{C_j}))$ (From (7.8))
$= min(\pi^t_{C_i}, \pi^t_{C_j}, max_{C_i \setminus S_{ij}} \pi^t_{C_i}, max_{C_j \setminus S_{ij}} \pi^t_{C_j})$
$= min(\pi^t_{C_i}, \pi^t_{C_j})$ (since $max_{C_i \setminus S_{ij}} \pi^t_{C_i} \geq \pi^t_{C_i}$ and $max_{C_j \setminus S_{ij}} \pi^t_{C_j} \geq \pi^t_{C_j}$).

Proof of Proposition 7.5 We want to prove that when the moral graph MG is one-neighbor stabilized then all the clusters have the same maximum value i.e. $\forall C_i, \max \pi^S_{C_i} = \alpha$.
Let C_i and C_j (s.t. $i < j$) be any two clusters in MG such that $max_{c_i} \pi^S_{C_i} = \alpha$, then we will show that $max_{c_j} \pi^S_{C_j} = \alpha$.
Between the two clusters C_i and C_j exists at least one path (otherwise they will be disconnected which is impossible by construction).

Let $A = C_i \cap C_{i+1}$ be the separator existing between C_i and C_{i+1}. Let c^m_i be one instance

of C_i having the maximum potential i.e. $\pi^S_{C_i}(c_i^m) = \alpha$ and let $a = c_i^m[A]$.

From the stability condition $\pi^S_{C_i}(c_i^m) = \alpha \Rightarrow \pi^S_A(a) = \alpha$.
Thus $\exists c_{i+1}$ s.t $c_{i+1}[A] = a$ and $\pi^S_{C_{i+1}}(c_{i+1}) = \alpha$ otherwise $max_{C_{i+1}\setminus A}\pi^S_{C_{i+1}}(a) < \alpha$ which contradicts the stability condition, this implies that $max_{c_{i+1}}\pi^S_{C_{i+1}}(c_{i+1}) = \alpha$
When iterating the same operation on $(C_{i+1}, ..., C_j)$, we deduce that $max\pi^S_{C_j} = \alpha$.

Proof of Proposition 7.7 Let $\pi^t_{\mathcal{MG}}$ be the joint distribution relative to a moral graph $\mathcal{MG}$ at level t. Let $\pi^{t+1}_{\mathcal{MG}}$ be the resulted joint distribution after the modification of C_i with respect to its two parents C_j and C_k using equation (7.18). We want to prove that $\pi^t_{\mathcal{MG}} = \pi^{t+1}_{\mathcal{MG}}$.
To show this proof it is enough to prove that $min(\pi^{t+1}_{C_i}, \pi^t_{C_j}, \pi^t_{C_k}) = min(\pi^t_{C_i}, \pi^t_{C_j}, \pi^t_{C_k})$. This relation is true, indeed:
$min(\pi^{t+1}_{C_i}, \pi^t_{C_j}, \pi^t_{C_k})$
$= min(min(\pi^t_{C_i}, \pi_S), \pi^t_{C_j}, \pi^t_{C_k})$ (From 7.18)
$= min(\pi^t_{C_i}, \pi_S, \pi^t_{C_j}, \pi^t_{C_k})$
$= min(\pi^t_{C_i}, \max_{C\setminus S} \pi^t_C, \pi^t_{C_j}, \pi^t_{C_k})$ (From (7.17))
$= min(\pi^t_{C_i}, \max_{C\setminus S} min(\pi^t_{C_j}, \pi^t_{C_k}), \pi^t_{C_j}, \pi^t_{C_k})$ (From (7.16))
$= min(\pi^t_{C_i}, min(\max_{C_j\setminus S_{ij}} \pi^t_{C_j}, \max_{C_k\setminus S_{ik}} \pi^t_{C_k}), \pi^t_{C_j}, \pi^t_{C_k})$
$= min(\pi^t_{C_i}, \max_{C_j\setminus S_{ij}} \pi^t_{C_j}, \max_{C_k\setminus S_{ik}} \pi^t_{C_k}, \pi^t_{C_j}, \pi^t_{C_k})$
$= min(\pi^t_{C_i}, \pi^t_{C_j}, \pi^t_{C_k})$ (since $\max_{C_j\setminus S_{ij}} \pi^t_{C_j} \geq \pi^t_{C_j}$ and $\max_{C_k\setminus S_{ik}} \pi^t_{C_k} \geq \pi^t_{C_k}$).

Proof of Lemma 7.1 Let $\mathcal{MG}$ be a moral graph and let $\pi_{\mathcal{MG}}$ be its joint distribution. We want to prove that if all the clusters of $\mathcal{MG}$ are consistent, then there exists an α-DAG $\mathcal{G}'$ such that its joint distribution π'_m is equal to $\pi_{\mathcal{MG}}$.
The α-DAG $\mathcal{G}'$ can be constructed from the moral graph $\mathcal{MG}$ by affecting a node A_i to each cluster C_i (s.t. $A_i \cup U_{A_i} \subseteq C_i$). Then, each link relative to A_i in $\mathcal{G}'$ is quantified by transforming the joint distribution relative to C_i into a conditional one.
This transformation is possible since any cluster C_i of $\mathcal{MG}$ is consistent, which means that for any instance u_{A_i} of U_{A_i} (where A_i is the variable relative to C_i),
$\max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \alpha$ (from 7.22).
Thus, when transforming the joint distribution relative to C_i into a conditional one, we obtain $\max_{a_i} \Pi(a_i \mid u_{A_i}) = \alpha$ which satisfies the normalization constraint of α-normalized min-based possibilistic networks. Moreover, using (7.4) and (5.6), respectively, we have:
- $\pi_{\mathcal{MG}}(A_1, .., A_N) = \min_{i=1..N} \pi_{C_i}(A_i \wedge U_{A_i})$ and
- $\pi'_m(A_1, .., A_N) = \min_{i=1..N} \Pi(A_i \mid U_{A_i})$.
Thus, we deduce that $\pi_{\mathcal{MG}} = \pi'_m$ since we consider that $\Pi(A_i \mid U_{A_i}) = \pi_{C_i}(A_i \wedge U_{A_i})$.

Proof of Proposition 7.10 We want to prove that a moral graph $\mathcal{MG}$ is consistent if all its clusters are consistent. This is equivalent to prove that $h(\pi_{\mathcal{MG}}) = \alpha$.
Let $\mathcal{MG}$ be an a moral graph such that all its clusters are consistent. Then, from Lemma 7.1, it exists an α-DAG $\mathcal{G}'$ such that its joint distribution π'_m satisfies $\pi_{\mathcal{MG}} = \pi'_m$. Moreover, from Proposition 7.1 we have $h(\pi'_m) = \alpha$. Thus, $h(\pi_{\mathcal{MG}}) = \alpha$.

Proof of Proposition 7.11 Let $\pi^t_{\mathcal{MG}}$ be the joint distribution relative to a moral graph $\mathcal{MG}$ at level t. Let $\pi^{t+1}_{\mathcal{MG}}$ be the resulted joint distribution obtained as result of the modification procedure. We want to prove that $\pi^t_{\mathcal{MG}} = \pi^{t+1}_{\mathcal{MG}}$.
Let C_i be an inconsistent cluster (relative to the variable A_i) and C_j be its parent cluster created in step 2.0 of the modification procedure.

To show the proof, it is enough to prove that for any particular instances c_i and c_j of of C_i and C_j, respectively, we satisfy: $min(\pi^t_{C_i}(c_i), \pi^t_{C_j}(c_j)) = min(\pi^{t+1}_{C_i}(c_i), \pi^{t+1}_{C_j}(c_j))$.
Let $u_{A_i} = c_i[U_{A_i}]$, we can distinguish three cases:

- $\max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \alpha$.

  In this case, $\pi^{t+1}_{C_i}(c_i) = \pi^t_{C_i}(c_i)$ and $\pi^{t+1}_{C_j}(c_j) = \pi^t_{C_j}(c_j)$ since the cluster is consistent with respect to u_{A_i}. Thus, the equality is trivial.

- $\max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \beta < \alpha$ and $\pi^t_{C_i}(c_i) < \beta$. Then:

  (i1) $\pi^{t+1}_{C_i}(c_i) = \pi^t_{C_i}(c_i) < \beta$ (since the modification will be performed on the potential c'_i s.t $\pi^t_{C_i}(c'_i) = \max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \beta$). Thus, two situations can occur:

  - $\pi^t_{C_j}(c_j) \leq \beta \Rightarrow \pi^{t+1}_{C_j}(c_j) = \pi^t_{C_j}(c_j)$. In this case, the equality is trivial.
  - $\pi^t_{C_j}(c_j) > \beta \Rightarrow \pi^t_{C_j}(c_j) > \pi^t_{C_i}(c_i)$ (since $\pi^t_{C_i}(c_i) < \beta$)
    $\Rightarrow$ (ii1) $min(\pi^t_{C_j}(c_j), \pi^t_{C_i}(c_i)) = \pi^t_{C_i}(c_i)$.
    Moreover, $\pi^t_{C_j}(c_j) > \beta$ implies that $\pi^{t+1}_{C_j}(c_j) = \beta$ (step 2.1 of the modification procedure). Thus, from (i1) and (ii1) we deduce that $min(\pi^t_{C_i}(c_i), \pi^t_{C_j}(c_j)) = min(\pi^{t+1}_{C_i}(c_i), \pi^{t+1}_{C_j}(c_j)) = \pi^t_{C_i}(c_i) = \pi^{t+1}_{C_i}(c_i)$.

- $\max_{a_i} \pi^t_{C_i}(a_i \wedge u_{A_i}) = \beta < \alpha$ and $\pi^t_{C_i}(c_i) = \beta$. Then:
  (i2) $\Rightarrow \pi^{t+1}_{C_i}(c_i) = \alpha > \pi^t_{C_i}(c_i)$.
  Thus, two situations can occur:

  - $\pi^t_{C_j}(c_j) \leq \beta \Rightarrow \pi^t_{C_j}(c_j) \leq \pi^t_{C_i}(c_i)$ (since $\pi^t_{C_i}(c_i) = \beta$)
    $\Rightarrow$ (ii2) $min(\pi^t_{C_j}(c_j), \pi^t_{C_i}(c_i)) = \pi^t_{C_j}(c_j)$.

Moreover, $\pi^t_{C_j}(c_j) \leq \beta$ implies that $\pi^{t+1}_{C_j}(c_j) = \pi^t_{C_j}(c_j)$ (step 2.1 of the modification procedure). Thus, from (i2) and (ii2) we deduce that $min(\pi^t_{C_i}(c_i), \pi^t_{C_j}(c_j)) = min(\pi^{t+1}_{C_i}(c_i), \pi^{t+1}_{C_j}(c_j)) = \pi^t_{C_j}(c_j) = \pi^{t+1}_{C_j}(c_j)$

– $\pi^t_{C_j}(c_j) > \beta \Rightarrow \pi^t_{C_j}(c_j) > \pi^t_{C_i}(c_i) = \beta$ (since $\pi^t_{C_i}(c_i) = \beta$)
$\Rightarrow min(\pi^t_{C_j}(c_j), \pi^t_{C_i}(c_i)) = \pi^t_{C_i}(c_i) = \beta$.
Moreover, $\pi^t_{C_j}(c_j) > \beta$ implies that $\pi^{t+1}_{C_j}(c_j) = \beta$ (step 2.1 of the modification procedure). Thus, $min(\pi^{t+1}_{C_i}(c_i), \pi^{t+1}_{C_j}(c_j)) = min(\pi^t_{C_j}(c_j), \pi^t_{C_i}(c_i)) = \beta$.

Appendix F

A Detailed Analysis of Procedures Used in the Anytime Propagation Algorithm

This appendix presents the main data structures and procedure used in the implementation of our anytime propagation algorithm studied in Chapter 7.

F.1 Data structures

The first step, in this propagation algorithm, is to create the possibilistic network called **pnet** and defined as follows:

pnet : Record **begin**
- **nodes**: 1-by-N matrix,
row vector containing nodes in a topological order (ancestors before descendants);
- **node_sizes**: 1-by-N matrix
node_sizes(i) is the number of values node i can take on (its arity);
- **dag**: N-by-N matrix
dag(i, j) = 1 if and only if i is parent of j;
- **CPD**: 1-by-N cell array of matrices
Each cell CPD{i} contains a *tabular_cpd object* defined by:
  - CPD{i}.self : Node i
  - CPD{i}.CPT : A vector containing the initial Conditional Possibility Distribution of node i in the context of its parents;

end

Once the the possibilistic network is defined, it is possible to create its moral graph and to perform propagation on it. To do so, we use a principle class: **MG_inf_engine** (*Moral Graph inference engine*) relative to the manipulation of moral graphs and to the global propagation.

Figure F.1: Multiply Connected DAG of Example F.1

An object of the class **MG_inf_engine** is called **engine** and is defined by:

engine : Record **begin**
- **- clusters**: C-by-C cell array of matrices,
clusters{i} corresponds to the nodes in cluster i;
- **- separators**: C-by-C cell array of matrices,
separator{i, j} corresponds to the nodes in the separator existing between clusters{i} and clusters{j};
- **- clpot**: 1-by-C cell array of matrices.
Each cell clpot{i} contains a *discrete potential object* (dpot) defined by:
  - ∗ clpot{i}.domain : 1-by-N matrix containing the variables existing in the cluster i,
  - ∗ clpot{i}.sizes : 1-by-N matrix containing the sizes of the variables existing in the cluster i,
  - ∗ clpot{i}.T: corresponds to the potential of cluster i.

end

The following example illustrates the values taken by different fields of the **pnet** and **engine** records.

Example F.1 *Let us consider the min-based possibilistic network ΠG_m treated in the previous Chapter. This ΠG_m composed by the DAG of Figure F.1 is quantified by the initial distributions given in Tables F.1 and F.2. The variable of evidence is D, its value is d_2, the variable of interest is A and the instance of interest is a_2.*

Table F.1: A priori and conditional possibilities

a	$\Pi(a)$	b	a	$\Pi(b \mid a)$	c	a	$\Pi(c \mid a)$
a_1	*1*	b_1	a_1	*1*	c_1	a_1	*0.3*
a_2	*0.9*	b_1	a_2	*0*	c_1	a_2	*1*
		b_2	a_1	*0.4*	c_2	a_1	*1*
		b_2	a_2	*1*	c_2	a_2	*0.2*

Table F.2: Conditional possibilities

d	b	c	$\Pi(d \mid b \wedge c)$	d	b	c	$\Pi(d \mid b \wedge c)$
d_1	b_1	c_1	*1*	d_2	b_1	c_1	*1*
d_1	b_1	c_2	*1*	d_2	b_1	c_2	*0*
d_1	b_2	c_1	*1*	d_2	b_2	c_1	*0.8*
d_1	b_2	c_2	*1*	d_2	b_2	c_2	*1*

To encode such possibilistic network, we should first define the topological order (i.e. ancestors before descendants) which can be $[ABCD]$ or $[ACBD]$. In the next, we will choose

$[ABCD]$ *i.e.* $A = 1, B = 2, C = 3$ *and* $D = 4$. *Then, the* ***pnet*** *record takes the following values:*

- ***pnet.nodes***= *[1 2 3 4]*
- ***pnet.node_sizes***= *[2 2 2 2]*
- ***pnet.dag***= $\begin{pmatrix} 0 & \boldsymbol{1} & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

  For instance the value 1 in the line 1 column 2 means that the variable 1 (A) is a parent of the variable 2 (B).

- *The conditional possibility distributions are stored as multidimentional arrays (pnet.CPD) where the variables are arranged s.t the low numbered parents come before the high numbered one:*

  pnet.CPD= *[1x1 tabular_cpd] [1x1 tabular_cpd] [1 x1 tabular_cpd] [1x 1 tabular_cpd]*

  - *pnet.CPD{1}.self = 1*

    pnet.CPD{1}.CPT = $\begin{pmatrix} 1 \\ 0.9 \end{pmatrix}$

    The order of instances in pnet.CPD{1}.CPT is as follows: $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$

    The value 1 (resp. 2) corresponds to the first (resp. second) instance of the variable 1 (A).
  - *pnet.CPD{2}.self = 2*

    pnet.CPD{2}.CPT = $\begin{pmatrix} 1 & 0.4 \\ 0 & 1 \end{pmatrix}$

    The order of instances in pnet.CPD{2}.CPT is as follows: $\begin{pmatrix} 11 & 12 \\ 21 & 22 \end{pmatrix}$

    The value 11 (resp. 21, 12, 22) corresponds to the first (resp. second, first, second) instance of the the variable 1 (A) and the first (resp. first, second, second) instance of the variable 2 (B).

- *pnet.CPD{3}.self = 3*

  $pnet.CPD\{3\}.CPT = \begin{pmatrix} 0.3 & 1 \\ 1 & 0.2 \end{pmatrix}$

  The order of instances in pnet.CPD{3}.CPT is as follows: $\begin{pmatrix} 11 & 12 \\ 21 & 22 \end{pmatrix}$

  The value 11 (resp. 21, 12, 22) corresponds to the first (resp. second, first, second) instance of the the variable 1 (A) and the first (resp. first, second, second) instance of the variable 3 (C).

- *pnet.CPD{4}.self = 4*

  $pnet.CPD\{4\}.CPT = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0.8 & 1 \end{pmatrix}$

  The order of instances in pnet.CPD{4}.CPT is as follows: $\begin{pmatrix} 111 & 121 & 112 & 122 \\ 211 & 221 & 212 & 222 \end{pmatrix}$

  The value 111 (resp. 211, 121, 221, 112, 212, 122, 222) corresponds to the first (resp. second, first, second, first, second, first, second) instance of the the variable 2 (B), the first (resp. first, second, second, first, first, second, second) instance of the variable 3 (C) and to the first (resp. first, first, first, second, second, second, second) instance of the variable 4 (D).

The moral graph corresponding to this possibilistic network is represented by the ***engine*** *record which takes the following values:*

- ***engine.clusters*** *= [1] [1x2 double] [1x2 double] [1x3 double]*
  engine.clusters{1} =[1] (i.e the cluster 1 contains the variable 1 (A))
  engine.clusters{2} =[1 2] (i.e the cluster 2 contains the variables 1 and 2 (AB))
  engine.clusters{3} =[1 3] (i.e the cluster 3 contains the variables 1 and 3 (AC))
  engine.clusters{4}=[2 3 4] (i.e the cluster 4 contains the variables 2, 3 and 4 (BCD))

- ***engine.separators*** $= \begin{pmatrix} [] & [\mathbf{1}] & [1] & [] \\ [] & [] & [] & [2] \\ [] & [] & [] & [3] \\ [] & [] & [] & [] \end{pmatrix}$

  For instance the value 1 in the line 1 column 2 means that there exists a separator containing the variable 1 (A) between the cluster 1 (A) and the cluster 2 (AB).

- ***engine.clpot***=*[1x1 dpot] [1x1 dpot] [1x1 dpot] [1x1 dpot]*

  - *engine.clpot{1}.domain: [1]*
    engine.clpot{1}.sizes: [2]
    engine.clpot{1}.T: $\begin{pmatrix} 1 \\ 0.9 \end{pmatrix}$
  - *engine.clpot{2}.domain= [1 2]*
    engine.clpot{2}.sizes: [2 2]
    engine.clpot{2}.T: $\begin{pmatrix} 1 & 0.4 \\ 0 & 1 \end{pmatrix}$
  - *engine.clpot{3}.domain: [1 3]*
    engine.clpot{3}.sizes: [2 2]
    engine.clpot{3}.T: $\begin{pmatrix} 0.3 & 1 \\ 1 & 0.2 \end{pmatrix}$
  - *engine.clpot{4}.domain: [2 3 4]*
    engine.clpot{4}.sizes: [2 2 2]
    engine.clpot{4}.T: $\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0.8 & 1 \end{pmatrix}$

  After incorporating the evidence $(D = d_2)$ *in this cluster it will be transformed into:*
  engine.clpot{4}.domain: [2 3 4]
  engine.clpot{4}.sizes: [2 2 2]
  engine.clpot{4}.T: $\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0.8 & 1 \end{pmatrix}$

F.2 Principle programs

In this Section we present the principle procedures for building possibilistic networks and moral graphs. Then, we give main procedures relative to the anytime propagation algorithm.

F.2.1 Building a possibilistic network structure

mk_pnet: Makes the possibilistic network structure (i.e. *pnet*) using the initial dag (i.e. *dag*), the node sizes (i.e. *node_sizes*) and the list of nodes (i.e. *nodes*).

Algorithm F.1: mk_pnet

```
Data: dag, node_sizes, nodes
Result: pnet
begin
    n ← length(dag);
    pnet.dag ← dag;
    pnet.node_sizes ← node_sizes(:)';
    pnet.nodes ← nodes;
end
```

F.2.2 Initialization procedure

- **MG_inf_engine**: (constructor of the class *MG_inf_engine*) creates a moral graph from the initial possibilistic network (i.e. *pnet*) composed of a cluster set (i.e. *engine.clusters*) using *dag_to_clusters* and a separator set (i.e. *engine.separators*). Each cluster i.e. *engine.clusters{i}* contains the node *i* and its parents and each separator i.e. *engine.separators{i, j}* contains the intersection of its relative clusters i.e. *engine.clusters{i}* and *engine.clusters{j}*.

Algorithm F.2: MG_inf_engine

```
Data: pnet
Result: engine
begin
    N ← length(pnet.dag);
    engine.clusters ← dag_to_clusters(pnet);
    num_clusters ← length(engine.clusters);
    for i←1 to num_clusters do
        for j←(i+1) to num_clusters do
            engine.separators{i, j} ← engine.clusters{i} ∩ engine.clusters{j};
end
```

- **dag_to_clusters**: generates the cluster set from the initial possibilistic network (i.e. *pnet*) by adding to each variable i its parent set (i.e. *ps*).

Algorithm F.3: dag_to_clusters

```
Data: pnet
Result: clusters
begin
    N ← length(pnet.dag);
    ns ← pnet.node_sizes(:);
    clusters ← {};
    j ← 1;
    for i ← 1 to N do
        ps ← parents(pnet.dag, i);
        clusters{j} ← ps ∪ {i};
        j ← j+1;
end
```

F.2.3 Global propagation

- **global_propagation**: Using the initial conditional distributions (i.e. *pnet.CPD*), this procedure first initializes the potentials of different clusters (i.e. *engine.clpot*) using *quantify_clusters* procedure. The propagation result depends on the evidence (i.e. *evidence*) and the instance of interest (i.e. *interest*). Indeed,

  - if the evidence and the instance of interest are specified, then this procedure provides the conditional possibility degree of the instance of interest in the context of evidence,
  - if only the evidence is specified, then this procedure provides the most plausible instances relative to all variables (except observed ones),
  - if only the instance of interest is specified, then this procedure provides the possibility degree of this instance (with no context),
  - if neither the evidence nor the instance of interest are specified, then this procedure provides the most plausible instances relative to all variables (given initial distributions).

  The global propagation uses the propagation parameters on consistency and stabilization procedures (i.e. *ck_cst*, *nodes_type*, *nb_nodes*) to call *enter_instance_propagate* procedure.

Algorithm F.4: global_propagation

```
Data: engine, interest, evidence, ck_cst, nb_nodes, nodes_type
Result: Poss_degree, best_instances
begin
    Poss_degree ← [];
    best_instances ← [];
    pnet ← pnet_from_engine(engine);
    ns ← pnet.node_sizes(:);
    N ← length(pnet.dag);
    CPDpot ← quantify_clusters(pnet, N);
    if evidence ≠ ∅ then
        if interest ≠ ∅ then
            [engine, pnet, Bel_evidence, clpot] ← enter_instance_propagate(engine, pnet,
            CPDpot, evidence, ck_cst, nb_nodes, nodes_type) ;
            [engine, pnet, Bel_joint_interest_evidence, clpot] ← enter_instance_propagate(engine, pnet, clpot, interest, ck_cst, nb_nodes,
            nodes_type) ;
            if Bel_joint_interest_evidence = Bel_evidence then
                Poss_degree ← 1;
            else
                Poss_degree ← Bel_joint_interest_evidence;
        else
            [engine, pnet, Poss_degree , clpot] ← enter_instance_propagate(engine, pnet,
            CPDpot, evidence, ck_cst, nb_nodes, nodes_type) ;
            best_instances ← define_best_instances(clpot, N, Poss_degree);
    else
        if interest ≠ ∅ then
            [engine, pnet, Poss_degree , clpot] ← enter_instance_propagate(engine, pnet,
            CPDpot, interest, ck_cst, nb_nodes, nodes_type) ;
        else
            [engine, pnet, Poss_degree, clpot] ← enter_instance_propagate(engine, pnet,
            CPDpot, evidence, ck_cst, nb_nodes, nodes_type) ;
            best_instances ← define_best_instances(clpot, N, Poss_degree);
end
```

- **enter_instance_propagate**: This procedure uses *incorporate_instance* to incorporate the evidence (i.e. *evidence*) or the instance of interest (i.e. *interest*) in its relative clusters. If we want to apply the consistency procedure by adding links (i.e. *ck_cst=1*) then, the propagation is performed via an iterative process of stabilization and checking consistency using *stabilization* and *consistency_by_adding_links*, successively until reaching the global consistency (i.e. *global_consistency =1*).

  If we want to apply the consistency procedure by testing the global instances (i.e. *ck_cst=2*) then, the propagation is performed via an iterative process of stabilization and consistency using *stabilization* and *consistency_by_computing_global_instances*,

successively until reaching the global consistency (i.e. *global_consistency =1*). Otherwise (i.e. *ck_cst=0*), we just stabilize the moral graph at the specified type of nodes (i.e. *nodes_type*) and number of nodes (i.e. *nb_nodes*).

Algorithm F.5: enter_instance_propagate

```
Data: engine, pnet, potential, instance, ck_cst, nb_nodes, nodes_type
Result: engine, pnet, Bel_instance, clpot
begin
    ns ← pnet.node_sizes(:);
    C ← length(engine.clusters);
    clpot ← potential;
    if instance ≠ ∅ then
        var_instance ← find(instance ≠ ∅);
        for i ← 1 to length(var_instance) do
            cl_var_instance ← engine.clq_ass_to_node(var_instance(i)) ;
            [clpot{cl_var_instance}] ← incorporate_instance (clpot{cl_var_instance},
            var_instance(i), instance{var_instance(i)}, ns);
    if ck_cst=1 then
        global_consistency ← 0;
        while global_consistency=0 do
            [clpot, alpha_stable] ← stabilize(pnet, engine, clpot, C, ns, nb_nodes, nodes_type);
            [engine, pnet, clpot, global_consistency]
            ← consistency_by_adding_links(engine, pnet, clpot, C, alpha_stable, ns);
        Bel_instance ← alpha_consistency;
    else
        if ck_cst=2 then
            global_consistency ← 0;
            while global_consistency=0 do
                [clpot, alpha_stable] ← stabilize(pnet, engine, clpot, C, ns, nb_nodes,
                nodes_type);
                [engine, clpot, global_consistency, alpha_consistency]
                ← consistency_by_computing_global_instances(engine, clpot, al-
                pha_stable, C);
            Bel_instance ← alpha_consistency;
        else
            [clpot, alpha_stable] ← stabilize(pnet, engine, clpot, C, ns, nb_nodes, nodes_type);
            Bel_instance ← alpha_stable;
    engine.clpot ← clpot;
end
```

F.2.4 Stabilization procedures

- **stabilize**: This procedure calls the different stabilization procedures depending on the type of nodes (i.e. *nodes_type* which can be parents, children, parents-children, neighbors) and number of nodes (i.e. *nb_nodes* where the value 1 (resp. 2, 3, n-best, n) is relative to simple (resp. two-nodes, three-nodes, n-best-nodes, n-nodes) stability).

Algorithm F.6: stabilize

```
Data: pnet, engine, clpot, C, ns, nb_nodes, nodes_type
Result: clpot, alpha_stable
begin
    switch nb_nodes do
        case 1
            [clpot] ← simple_stability(engine, pnet, clpot, C);
            alpha_stable ← maximum_value(clpot{1});
        case 2
            modif_pot←1;
            while modif_pot=1 do
                [clpot] ← simple_stability(engine, pnet, clpot, C);
                [clpot, modif_pot] ← two_nodes_stability(pnet, engine, clpot, C, ns,
                nodes_type);
            alpha_stable ← maximum_value(clpot{1});
        case 3
            modif_pot←1;
            while modif_pot=1 do
                [clpot] ← simple_stability(engine, pnet, clpot, C);
                [clpot, modif_pot] ← three_nodes_stability(pnet, engine, clpot, C, ns,
                nodes_type);
            alpha_stable ← maximum_value(clpot{1});
        case n-best
            modif_pot←1;
            while modif_pot=1 do
                [clpot] ← simple_stability(engine, pnet, clpot, C);
                [clpot, modif_pot]← best_multiple_nodes_stability (pnet, engine, clpot, C,
                ns, nodes_type);
            alpha_stable ← maximum_value(clpot{1});
        case n
            modif_pot←1;
            while modif_pot=1 do
                [clpot] ← simple_stability(engine, pnet, clpot, C);
                [clpot, modif_pot] ← multiple_nodes_stability (pnet, engine, clpot, C, ns,
                nodes_type);
            alpha_stable ← maximum_value(clpot{1});
end
```

- **simple_stability**: This procedure ensures that any cluster agrees with each of its neighbors on the distributions defined on common variables. Let $d = (C_1, ..., C_N)$ be any ordering of the clusters such that $U_{A_i} \subseteq \{A_1, ..A_{i-1}\}$. In the implementation proposed in Appendix F, the order in which messages circulate during the simple stability procedure depends on d. Indeed, we start with the last cluster in d which is stabilized w.r.t. of all its neighbors, then its predecessor will be treated and it will be stabilized w.r.t. of all its neighbors except those who already use it. A cycle is achieved when all the clusters are treated. This process will be repeated until reaching the stability (i.e *stable*=1).

Algorithm F.7: simple_stability

```
Data: engine, pnet, clpot, C
Result: clpot
begin
    stable ← 0;
    while stable = 0 do
        for i ← C down to 1 do
            for j ← i-1 down to 1 do
                if engine.separators{j, i} ≠ ∅ then
                    seppot{j, i} ← marginalize_pot(clpot{i}, engine.separators{j, i});
                    sauvpot{j, i} ← seppot{j, i};
                    seppot{j, i} ← marginalize_pot(clpot{j}, engine.separators{j, i});
                    seppot{j, i} ← minimize_by_pot(sauvpot{j, i}, seppot{j, i});
                    clpot{i} ← minimize_by_pot(clpot{i}, seppot{j, i});
                    clpot{j} ← minimize_by_pot(clpot{j}, seppot{j, i});
        test ← 1;
        i ← 1;
        while (i ≤ C) and (test = 1) do
            for j ← i+1 to C do
                if engine.separators{j, i} ≠ ∅ then
                    seppot{i, j} ← marginalize_pot(clpot{i}, engine.separators{i, j});
                    sauvpot{i, j} ← seppot{i, j};
                    seppot{i, j} ← marginalize_pot(clpot{j}, engine.separators{i, j});
                    equal ← test_equality(sauvpot{i, j}, seppot{i, j});
                    if equal = 0 then test ← 0;
                j ← j+1;
            i ← i+1;
        if test = 1 then stable ← 1;
end
```

- **two_nodes_stability**: This procedure ensures for each cluster having at least two nodes (i.e. *nodes_type* which can be neighbors, parents, children or parents-children) its stability with respect to each pair of them using the procedure *newpot_n_nodes*.

Algorithm F.8: two_nodes_stability

```
Data: pnet, engine, clpot, C, ns, nodes_type
Result: clpot, modif_pot
begin
    modif_pot ← 0;
    i ← 1;
    while (i ≤ C) and (modif_pot = 0) do
        nodes ← define_nodes(pnet.dag, engine, i, nodes_type, C);
        if length(nodes) ≥ 2 then
            j ← 1;
            while j ≤ length(nodes)-1 and modif_pot=0 do
                first_node ← nodes(j);
                k ← 1;
                while k ≤ length(nodes) and modif_pot=0 do
                    second_node ← nodes(k);
                    two_nodes ← first_node ∪ second_node;
                    [clpot{i}, modif_pot] ← newpot_n_nodes(clpot{i}, clpot,
                    two_nodes, ns);
                    k ← k+1;
                j ← j+1;
        i ← i+1;
end
```

- **three_nodes_stability**: This procedure ensures for each cluster having at least three nodes (i.e. *nodes_type* which can be neighbors, parents, children or parents-children) its stability with respect to each pair of them using the procedure *newpot_n_nodes*.

Algorithm F.9: three_nodes_stability

```
Data: pnet, engine, clpot, C, ns, nodes_type
Result: clpot, modif_pot
begin
    modif_pot ← 0;
    i ← 1;
    while (i ≤ C) and (modif_pot = 0) do
        nodes ← define_nodes(pnet.dag, engine, i, nodes_type, C);
        if length(nodes) ≥ 3 then
            j ← 1;
            while j ≤ length(nodes)-2 and modif_pot=0 do
                first_node ← nodes(j);
                k ← j+1;
                while k ≤length(nodes)-1 and modif_pot=0 do
                    second_node=nodes(k);
                    k ← k+1;
                    while l ≤ length(nodes) and modif_pot=0 do
                        third_node=nodes(l);
                        three_nodes= first_node ∪ second_node ∪ third_node;
                        [clpot{i}, modif_pot] ← newpot_n_nodes(clpot{i}, clpot,
                        three_nodes, ns);
                        l ← l+1;
                    k ← k+1;
                j ← j+1;
        else
            if length(nodes) =2 then
                j ← 1;
                while j ≤ length(nodes)-1 and modif_pot=0 do
                    first_node ← nodes(j);
                    k ← 1;
                    while k ≤ length(nodes) and modif_pot=0 do
                        second_node ← nodes(k);
                        two_nodes ← first_node ∪ second_node;
                        [clpot{i}, modif_pot] ← newpot_n_nodes(clpot{i}, clpot,
                        two_nodes, ns);
                        k ← k+1;
                    j ← j+1;
        i ← i+1;
end
```

- **multiple_nodes_stability**: This procedure ensures for each cluster its stability with respect to all its nodes (i.e. *nodes_type* which can be parents, children, parents-children or neighbors) using the *newpot_n_nodes* procedure.

Algorithm F.10: multiple_nodes_stability

```
Data: pnet, engine, clpot, C, ns, nodes_type
Result: clpot, modif_pot, cap_max
begin
    modif_pot ← 0;
    i← 1;
    while (i≤C) and (modif_pot=0) do
        nodes←define_nodes(pnet.dag, engine, i, nodes_type, C);
        if length(nodes)>1 then
            [clpot{i}, modif_pot, cap_max] ← newpot_n_nodes(clpot{i}, clpot, nodes,
            ns);
    i←i+1;
end
```

- **define_nodes**: Using the type of nodes (i.e. *nodes_type*), this procedure defines the node set which can be neighbors, parents, children or parents-children.

Algorithm F.11: define_nodes

```
Data: adj_mat, engine, i, nodes_type, C
Result: nodes
begin
    ps←parents(adj_mat, i);
    cs←children(adj_mat, i);
    switch nodes_type do
        case 'parents' nodes← ps
        case 'children' nodes← cs
        case 'parents-children' nodes← ps ∪ cs
        case 'neighbors'
            nodes← [];
            for j ← 1 to C do
                if j ≠ i then
                    if engine.clusters{i} ∩ engine.clusters{j} ≠ ∅ then
                        nodes← nodes ∪ j;
end
```

- **newpot_n_nodes**: This procedure modifies the potential of a cluster (i.e. *potcl*) using the cartesian product of some of its nodes which can be parents, children, parents-children or neighbors (i.e. *nodes*). The main steps of this procedure are:
  - compute the domain relative to variables in *nodes* (i.e. *big_domain*),
  - compute the potential relative to *big_domain* (i.e. *potential*),
  - compute the domain of the separators between the treated cluster and *nodes* (i.e *onto*),
  - marginalize the potential of the clusters to the separators (i.e. *inter*),
  - compute the new potential of the cluster using *inter* (i.e. *potcl*),
  - compare the initial potential of the cluster and the new one. If it is modified, then *modif_pot* takes the value 1.

Algorithm F.12: newpot_n_nodes

```
Data: potcl, clpot, nodes, ns
Result: potcl, modif_pot
begin
    save_pot ← potcl.T;
    big_domain ← [];
    for i ← 1 to length(nodes) do
        big_domain ← big_domain ∪ clpot{nodes(i)}.domain;
    p_inter ← dpot(big_domain, ns(big_domain));
    potential ← p_inter;
    for i ← 1 to length(nodes) do
        p_inter.T ← extend_domain_table(clpot{nodes(i)}.T, clpot{nodes(i)}.domain,
        clpot{nodes(i)}.sizes, p_inter.domain, p_inter.sizes);
        potential.T ← min(potential.T, p_inter.T);
    onto ← [];
    for i ← 1 to length(nodes) do
        onto ← onto ∪ (potcl.domain ∩ clpot{nodes(i)}.domain);
    inter ← marginalize_pot(potential, onto);
    potcl ← minimize_by_pot(potcl, inter);
    modif_pot ← 0;
    if save_pot ≠ potcl.T then modif_pot ← 1;
end
```

- **best_multiple_nodes_stability** : This procedure ensures for each cluster its stability with respect to the best instances of its nodes (which can be parents, children, parents-children or neighbors) using the procedure *newpot_n_best_nodes*.

Algorithm F.13: best_multiple_nodes_stability

```
Data: pnet, engine, clpot, C, ns, nodes_type
Result: clpot, modif_pot
begin
    modif_pot ← 0;
    i ← 1;
    while (i ≤ C) and (modif_pot = 0) do
        nodes ← define_nodes(pnet.dag, engine, i, nodes_type, C);
        if length(nodes)>1 then
            [clpot{i}, modif_pot] ← newpot_n_best_nodes(clpot{i}, clpot, nodes,
            ns);
        i ← i+1;
end
```

- **newpot_n_best_nodes**: This procedure updates the potential of a cluster using the best instances in the cartesian product relative to its nodes (i.e. *nodes*) which can be parents, children, parents-children or neighbors. The main steps of this procedure are:
  - compute the best instances in each node (i.e. *sauv_max_index*) and the order in which we should cover them (i.e. *clusters_order*) using *extract_best_instances*,
  - compute the domain relative to variables in *nodes* (i.e. *big_domain*),
  - compute the best instances (i.e. having the maximal degree) in the cartesian product relative to *big_domain* (i.e. *best_nodes_instances*) using *compute_best_nodes_instances*.
  - compute the domain of the separators between the treated cluster and *nodes* (i.e *onto*),
  - compute best instances relative to *onto* from *best_nodes_instances* (i.e *sep_instances_from_nodes*) using *extract_onto*,
  - compute best instances relative to *onto* from *best_cl_instances* (i.e *sep_instances_from_cluster*) using *extract_onto*,
  - test the coherence between *sep_instances_from_nodes* and *sep_instances_from_cluster*,
  - if incoherence (i.e *uncoherent_instances* $\neq \emptyset$), then we should decrease the degree of incoherent instances by choosing the next degree in *scale*.

Algorithm F.14: newpot_n_best_nodes

```
Data: potcl, clpot, nodes, ns
Result: potcl, modif_pot
begin
    modif_pot ← 0;
    scale ← define_scale(clpot, nodes);
    pos_val_in_scale ← length(scale);
    big_domain ← [];
    for i ← 1 to length(nodes) do big_domain ← big_domain ∪ clpot{nodes(i)}.domain);
    val_max_nodes← scale(pos_val_in_scale);
    [sauv_max_index, clusters_order ] ← extract_best_instances(potcl, val_max_nodes,
    clpot, nodes);
    cap_max ← 0;
    [best_nodes_instances] ←
    compute_best_nodes_instances(sauv_max_index, big_domain, potcl, clpot, nodes,
    clusters_order );
    onto ← [];
    for i ← 1 to length(nodes) do
        onto ← onto ∪ (potcl.domain ∩ clpot{nodes(i)}.domain);
    sep_instances_from_nodes ← extract_onto(potcl, onto, big_domain,
    best_nodes_instances);
    [best_cl_instances] ← extract_best_instances(potcl, val_max_cluster);
    sep_instances_from_cluster ← extract_onto(potcl, onto, potcl.domain, best_cl_instances,
    'val+pos');
    pos_incoherent_instance ← 1;
    incoherent_instances ← [];
    for i ← 1 to length(sep_instances_from_cluster) do
        j ← 1;
        my_test ← 0;
        while j ≤ length(sep_instances_from_nodes) and my_test = 0 do
            if sep_instances_from_nodes{j} = sep_instances_from_cluster {i}.val then
                my_test ← 1;
            j ← j+1;
        if my_test = 0 then
            incoherent_instances(pos_incoherent_instance) ←
            sep_instances_from_cluster {i}.pos;
            pos_incoherent_instance ← pos_incoherent_instance+1;
    if incoherent_instances = ∅ then
        x ← find(scale = val_max_nodes);
        if x ≠1 then
            potcl.T(incoherent_instances) ← scale(x-1);
            modif_pot ← 1;
end
```

F.2.5 Consistency procedures

As described in Section 7.10, the consistency can be applied via two procedures. The first one (i.e. *consistency_by_computing_global_instances*) checks the existence of a global instance by constructing the best elements in the cartesian product using *compute_best_global_instances*. The second procedure (i.e. *consistency_by_adding_links*) ensures the consistency by computing from each cluster the potential of its parents.

- **consistency_by_computing_global_instances**: This procedure tests the existence of a global consistency using *compute_best_global_instances*. If such instance exists (i.e. *exists_global_instance*=1), then the moral graph is consistent.
  Otherwise, (i.e. *exists_global_instance*=0) we should decrease the maximal value in the clusters using *decrease_in_scale*.

Algorithm F.15: consistency_by_computing_global_instances

```
Data: engine, clpot, alpha_stable, C
Result: engine, clpot, global_consistency, alpha_consistency
begin
    exists_global_instance ← 0;
    [exists_global_instance, scale, sauv_index_clusters] ←
    compute_best_global_instances(alpha_stable, clpot, C);
    if exists_global_instance=0 then
        for i ← 1 to C do
            clpot{i} ← decrease_in_scale(scale, alpha_stable, clpot{i},
            sauv_index_clusters{i});
        global_consistency ← 0;
    else
        global_consistency ← 1;
    alpha_consistency ← maximum_value(clpot{1}, C);
end
```

Algorithm F.16: compute_best_global_instances

```
Data: val_max, clpot, C
Result: exist_global_instance, scale, sauv_index_clusters, cap_max
begin
    scale ← []; exist_global_instance ← 1; treated_var ← [];
    best_global_instances ← []; big_domain ← [];
    [sauv_max_index, clusters_order ] ← extract_best_instances(C, val_max, clpot);
    for i ← 1 to C do  big_domain ← big_domain ∪ clpot{i}.domain;
    pos_treated_cluster ← C;
    first_cluster ← clusters_order (pos_treated_cluster);
    for i ← 1 to length(sauv_max_index{first_cluster}) do
        best_global_instances{i} ← zeros(1, length(big_domain));
    treated_cluster ← clpot{first_cluster};
    dom ← treated_cluster.domain;
    equiv_pos_dom ← find_equiv_posns(dom, big_domain);
    for i ← 1 to length(sauv_max_index{first_cluster}) do
        best_global_instances{i}([equiv_pos_dom]) ← [sauv_max_index{first_cluster}{i}];
    treated_var ← treated_var ∪ dom;
    scale ← update_scale(scale, treated_cluster);
    if C>2 then
        next ← 1;
        pos_treated_cluster ← 2;
        while (pos_treated_cluster ≤ C) and (next = 1) do
            next_cluster ← clusters_order (pos_treated_cluster);
            treated_cluster ← clpot{next_cluster};
            treated_coherent ← 0;
            dom ← treated_cluster.domain;
            equiv_pos_dom ← find_equiv_posns(dom, big_domain);
            if dom ∩ treated_var ≠ ∅  then  treated ← 0;
            else  treated ← 1;
            treated_var ← treated_var ∪ dom;
            scale ← update_scale(scale, treated_cluster);
            position_big_instance ← 1;
            sauv_best_global_instances ← best_global_instances;
            nb_best_global_instances ← length(best_global_instances)
            best_global_instances ← [];
            val_max_clusters ← sauv_max_index{next_cluster};
            length_val_max_clusters ← length(val_max_clusters);
            for i ← 1 to length_val_max_clusters  do
                one_cluster_instance ← sauv_max_index{next_cluster}{i}
                for j ← 1 to nb_best_global_instances do
                    if treated =1 then
                        one_big_instance ← sauv_best_global_instances{j};
                        test_coherence  ←  test_coherence_instance(one_cluster_instance,
                        one_big_instance, equiv_pos_dom);
                    if (treated=0) or (test_coherence = 1) then
                        treated_coherent ← 1;
                        best_global_instances{position_big_instance}
                        ← sauv_best_global_instances{j};
                        best_global_instances{position_big_instance}([equiv_pos_dom])
                        ← [sauv_max_index{next_cluster}{j}];
                        position_big_instance ← position_big_instance+1;
            if treated_coherent = 0 then
                next ← 0; exist_global_instance ← 0; scale ← sort(scale);
            pos_treated_cluster ← pos_treated_cluster+1;
end
```

- **consistency_by_adding_links**: This procedure ensures the global consistency by computing from each cluster having at least one parent the potential of its parents (i.e. *pot_parents*).
  Then, we check if this distribution contains some values less than *alpha-stable* using *check_consistency_cluster*. If so, then the cluster is inconsistent. In this case we should first test the following situations before dropping its inconsistency:

  – if the degree β exists in the parents of C_i (this test is performed using *check_beta*),

  – if the parents of the treated cluster are already linked in the DAG (i.e. *all_linked*=0) (from the construction or from additional links of a previous step).

  If none of these cases is true, then we should modify the inconsistent cluster as follows:

  1. Modification of the potential of the inconsistent cluster (replace *beta* by *alpha-stable* using *modify_pot*)
  2. Retrieval of *beta* by adding links between parents of i (i.e. *ps*) using *add_links*. To do so, we choose the parent in the maximum position (i.e. *parent_index*) and transform the other parents (i.e. *rest_parents*) as its parents in its corresponding cluster (i.e. q) so that to respect the topological order.

  Note that this consistency procedure starts from the leaves clusters until reaching the roots (i.e. from cluster number C down to number 1).

Algorithm F.17: consistency_by_adding_links

```
Data: engine, pnet, clpot, C, alpha_stable, ns
Result: engine, pnet, clpot, global_consistency
begin
    global_consistency ← 1;
    i ← C;
    while (global_consistency = 1) and (i ≥ 1) do
        ps ← parents(pnet.dag, i);
        if length(ps) > 1 then
            pot_parents ← marginalize_pot(clpot{i}, ps);
            consistency ← check_consistency_cluster(C, pot_parents, alpha_stable);
            if (consistency = 0) then
                modif_pot ← 0;
                test_beta ← 1;
                test_beta ← check_beta(pot_parents, clpot, ns, ps, alpha_stable);
                if test_beta=0 then
                    parent_index ← max_position(ps);
                    p ← ps(parent_index);
                    q ← engine.clq_ass_to_node(p);
                    rest_parents ← ps - p;
                    pp ← parents(pnet.dag, p);
                    all_linked ← 0;
                    if rest_parents ⊆ pp then all_linked ← 1;
                    if all_linked = 0 then
                        clpot{i} ← modify_pot(clpot{i}, ps, alpha_stable, ns);
                        [engine, pnet, clpot, modif_pot] ← add_links(engine, pnet, ns, ps,
                        clpot, p, q, parent_index, pot_parents, alpha_stable);
                        if (modif_pot = 1) then global_consistency ← 0;
        i ← i-1;
    alpha_consistency ← maximum_value(clpot{1});
end
```

Bibliography

[1] S. Arnborg, D. G. Corneil, and A. Prokurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal Algebraic and Discrete Methods*, 8:277–284, 1987.

[2] N. Ben Amor, S. Benferhat, D. Dubois, H. Geffner, and H. Prade. Independence in qualitative uncertainty frameworks. In *Proceedings of Principles of Knowledge Representation and Reasoning (KR'2000)*, pages 235–246, Colorado (USA), 2000.

[3] N. Ben Amor, S. Benferhat, D. Dubois, H. Geffner, and H. Prade. A qualitative approach to independence in reasoning under uncertainty. In *Proceedings of Workshop on Causal Independence Structures and Graphical Models*, Toronto (Canada), 2000.

[4] N. Ben Amor, S. Benferhat, D. Dubois, H. Geffner, and H. Prade. Une approche qualitative de l'indépendance pour le raisonnement dans l'incertain. In *Proceedings of Congré de la Reconnaissance des Formes et Intelligence Artificielle (RFIA'2000)*, pages 505–514, Paris (France), 2000.

[5] N. Ben Amor, S. Benferhat, D. Dubois, K. Mellouli, and H. Prade. A theoretical framework for possibilistic independence in a weakly ordered setting. *To appear in International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.

[6] N. Ben Amor, S. Benferhat, and K. Mellouli. Possibilistic independence vs qualitative independence. In *Proceedings of International Conference on Information Processing of Uncertainty in Knowledge Based Systems (IPMU'2000)*, pages 349–356, Madrid (Spain), 2000.

[7] N. Ben Amor, S. Benferhat, and K. Mellouli. Qualitative possibilistic independence based on plausibility relations. In B. Bouchon-Meunier, J. Gutierrez-Rios, L. Magdalena, and R.R. Yager, editors, *Technologies for Constructing Intelligent Systems*. Springer Verlag, 2000.

[8] N. Ben Amor, S. Benferhat, and K. Mellouli. A new propagation algorithm for min-based possibilistic causal networks. In *Proceedings of European Conference of Symbolic*

and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'2001), pages 266–277, Toulouse (France), 2001.

[9] N. Ben Amor, S. Benferhat, and K. Mellouli. Some issues on the coherence of min-based possibilistic causal networks. In *Proceedings of International Fuzzy Systems Association (IFSA'2001)*, pages 604–609, Vancouver (Canada), 2001.

[10] N. Ben Amor, S. Benferhat, and K. Mellouli. Anytime possibilistic propagation algorithm. In *Proceedings of the First International Conference on Computing in an Imperfect World (Soft-Ware'2002)*, pages 263–279, Belfast (Irland), 2002.

[11] N. Ben Amor, S. Benferhat, and K. Mellouli. Un algorithme de propagation pour les réseaux possibilistes basés sur le conditionnement ordinal. In *Proceedings of Congré de la Reconnaissance des Formes et Intelligence Artificielle (RFIA'2002)*, pages 339–348, Angers (France), 2002.

[12] B. Ben Yaghlane, P. Smets, and K. Mellouli. Belief function independence: I. the marginal case. *International Journal on Approximate Reasoning*, 29, 2002.

[13] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proceedings of Intlernational Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 640–645, Chambéry (France), 1993.

[14] S. Benferhat, D.Dubois, and H. Prade. Expressing independence in possibilistic framework and its application to default reasoning. In *Proceedings of European Conference on Artificial Intelligence (ECAI'94)*, pages 150–153, Amsterdam, 1994.

[15] S. Benferhat, D.Dubois, and H. Prade. Possibilistic independance and plausible reasoning. In *Proceedings of Foundations and Applications of Possibility Theory (FAPT'95)*, pages 47–63, Ghent (Belgique), 1995.

[16] S. Benferhat, D. Dubois, L. Garcia, and H. Prade. Possibilistic logic bases and possibilistic graphs. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99)*, pages 57–64, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[17] S. Benferhat, D. Dubois, L. Garcia, and H. Prade. On the transformation between possibilistic logic bases and possibilistic causal networks. *International Journal on Approximate Reasoning*, 29, 2002.

[18] S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Graphical readings of possibilistic logic bases. In *Proceedings of European Conference of Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'2001)*, pages 422–431, Toulouse (France), 2001.

[19] S. Benferhat, D. Dubois, and H. Prade. Nonmonotonic reasoning, conditional objects and possibility theory. *Artificial Intelligence*, 92:259–276, 1997.

[20] S. Benferhat, D. Dubois, H. Prade, and M. Williams. A practical approach to fusing and revising prioritized belief bases. In *Proceedings of EPIA'99*, pages 222–236, 1999.

[21] S. Benferhat and C. Sossai. Merging uncertain knowledge bases in a possibilistic logic framework. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 8–15, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

[22] C. Borgelt, J. Gebhardt, and R. Kruse. Inference methods. In *Handbook on Fuzzy Computation*, chapter F1.2, pages F1.2:1–F.1.2:13. E. Ruspini and P. Bonissone and W. Pedrycz, Institute of Physics Publishing, Bristol, United Kingdom, 1998.

[23] C. Borgelt, J. Gebhardt, and Rudolf Kruse. Possibilistic graphical models. In *Proceedings of International School for the Synthesis of Expert Knowledge (ISSEK'98)*, Udine (Italy), 1998.

[24] C. Borgelt and R. Kruse. Some experimental results on learning probabilistic and possibilistic networks with different evaluation measures. In *Proceedings of European Conference of Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU/FAPR'97)*, pages 71–85, Bad Honnef (Germany), 1997.

[25] R. R. Bouckaert. Bayesian belief networks and conditional independencies. In *Technical report RUU-CS-92-36*. Department of computer science, Utrecht University, 1992.

[26] C. Boutilier. Revision sequences and nested conditionals. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 519–531, Chambéry (France), 1993.

[27] C. Boutilier, R. I. Brafman, H. H. Hoos, and D. Poole. Reasoning with conditional ceteris paribus preference statements. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99)*, pages 71–80, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[28] J. Cano, M. Delgado, and S. Moral. An axiomatic framework for propagating uncertainty in directed acyclic networks. *International Journal of Approximate Reasoning*, 8:253–280, 1993.

[29] R. M. Chavez and G. F. Cooper. A randomized approximation algorithm for probabilistic inference on bayesian belief networks. *Networks*, 20:661–685, 1990.

[30] G. F. Cooper. Computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[31] F. G. Cozman. Irrelevance and independence axioms in quasi-bayesian theory. In *Proceedings of European Conference of Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'1999)*, pages 128–139, London, 1999.

[32] P. Dagum and M. Luby. Approximating probabilistic inference in bayesian belief networks is np-hard. *Artificial Intelligence*, 60(1):141–153, 1993.

[33] A. Darwiche. A logical notion of conditional independence: properties and applications. *Artificial Intelligence*, 97:45–82, 1997.

[34] A. Darwiche. A differential approach to inference in bayesian networks. In *Proceedings of Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI'2000)*, pages 123–132, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[35] A. Darwiche and J. Pearl. Symbolic causal networks. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI'94)*, pages 238–244, Seattle (Washington), 1994.

[36] A. Darwiche and J. Pearl. On the logic of iterated revision. *Artificial Intelligence*, 89:1–29, 1997.

[37] A. D. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society*, 41:1–31, 1979.

[38] L.M de Campos and J. F. Huete. Independence concepts in possibility theory: Part i. *Fuzzy Sets and Systems*, 103:127–152, 1999.

[39] L.M de Campos and J. F. Huete. Independence concepts in possibility theory: Part ii. *Fuzzy Sets and Systems*, 103:487–505, 1999.

[40] G. de Cooman and E. E. Kerre. A new approach to possibilistic independence. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, pages 1446–1451, 1994.

[41] L. F. del Cerro and A. Herzig. Possibility theory and independence. In *Proceedings of International Conference on Information Processing of Uncertainty in Knowledge Based Systems (IPMU'94)*, pages 512–517, Paris (France), 1994.

[42] J. Doyle and M. P. Wellman. Preferential semantics for goals. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI'91)*, pages 698–703, Seattle (Washington), 1991.

[43] D. Dubois. Belief structures, possibility theory and decomposable confidence measures on finite sets. *Computers and Artificial Intelligence*, 5:403–416, 1986.

[44] D. Dubois, D. Cayrac, and H. Prade. Handling uncertainty with possibility theory and fuzzy sets in a satellite fault diagnosis application. In *Proceedings of International Conference on Information Processing of Uncertainty in Knowledge Based Systems (IPMU'96)*, pages 251–269, Granada (Spain), 1996.

[45] D. Dubois, L. Farinas del Cerro, A. Herzig, and H. Prade. Qualitative relevance and independence: a roadmap. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'97)*, pages 62–67, Nagoya (Japon), 1997.

[46] D. Dubois, L.F. del Cerro, A. Herzig, and H. Prade. An ordinal view of independence with application to plausible reasoning. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI'94)*, pages 195–203, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[47] D. Dubois, H. Fargier, and H. Prade. Refinements of the maximin approach to decision-making in fuzzy environment. *Fuzzy Sets and Systems*, 81:103–122, 1996.

[48] D. Dubois, H. Fargier, and H. Prade. Comparative uncertainty, belief functions and accepted beliefs. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 113–120, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

[49] D. Dubois and P. Fortemps. Computing improved optimal solutions to max-min flexible constraint satisfaction problems. *European Journal of Operational Research*, 118:95–126, 1999.

[50] D. Dubois, P. Fortemps, M. Pirlot, and H. Prade. Leximin optimality and fuzzy set-theoretic operations. *European Journal of Operational Research*, 130:20–28, 2001.

[51] D. Dubois, L. Godo, H. Prade, and A. Zapico. Making decision in a qualitative setting: from decision under uncertainty to case-based decision. In *Proceedings of Principles of Knowledge Representation and Reasoning (KR'98)*, pages 594–607, Trento (Italie), 1998.

[52] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In *Handbook on Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University press, 1994.

[53] D. Dubois, S. Moral, and H. Prade. Belief change rules in ordinal and numerical uncertainty theories. In *Belief Change*, pages 311–392. Kluwer, Dordrecht, Pays-Bas, 1998.

[54] D. Dubois and H. Prade. *Possibility theory: An approach to computerized, Processing of uncertainty*. Plenium Press, New York, 1988.

[55] D. Dubois and H. Prade. Inference in probabilistic hypergraphs. In *Proceedings of International Conference on Information Processing of Uncertainty in Knowledge Based Systems (IPMU'90)*, pages 228–230, Paris (France), 1990.

[56] D. Dubois and H. Prade. Epistemic entrenchment and possibilistic logic. *Artificial Intelligence*, 50:223–239, 1991.

[57] D. Dubois and H. Prade. Possibility theory and data fusion in poorly informed environnment. *Control Ingeneering Practice*, 2:811–823, 1994.

[58] D. Dubois and H. Prade. Numerical representation of acceptance. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 149–156, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[59] D. Dubois and H. Prade. Possibility theory as a basis of qualitative decision theory. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1925–1930, 1995.

[60] D. Dubois and H. Prade. Possibility theory: qualitative and quantitative aspect. In D. Gabbay and P. Smets, editors, *Handbook on Defeasible Reasoning and Uncertainty Management Systems*, volume 1, pages 21–42. Jürg Kohlas and Serafin Moral, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[61] D. Dubois, H. Prade, and Ph. Smets. Representing partial ignorance. *IEEE System Machine and Cybernetic*, 26:361–377, 1996.

[62] P. Fonck. Conditional independence in possibility theory. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI'94)*, pages 221–226, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[63] P. Fonck. *Réseaux d'inférence pour le raisonnement possibiliste*. PhD thesis, Université de Liège, Faculté des Sciences, 1994.

[64] P. Fonck. A comparative study of possibilistic conditional independence and lack of interaction. *International Journal of Approximate Reasoning*, 16:149–171, 1997.

[65] S. French. *Decision theory: An introduction to the mathematics of rationality*. Jhon Wiley and Sons, New York, 1984.

[66] N. Friedman and J. Halpern. Plausible measures and default reasoning. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI'96)*, pages 1297–1304, Portland (Oregon), 1996.

[67] P. Gärdenfors. *Knowledge in flux: modeling the dynamics of epistemic states*. MIT Press, Cambridge, 1988.

[68] J. Gebhardt and R. Kruse. Possinfer - a software tool for possibilistic inference. In *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications*. Didier Dubois and Henri Prade and Ronald R. Yager, Jhon Wiley and Sons, New York, 1995.

[69] J. Gebhardt and R. Kruse. Background and perspectives of possibilistic graphical models. In *Proceedings of European Conference of Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'1997)*, pages 108–121, Bad Honnef (Germany), 1997.

[70] D. Geiger and A. Shaffer. Automatic selection of loops breakers for genetic linkage analysis. *Human Heredity*, 48, 1998.

[71] D. Geiger, T. Verma, and J. Pearl. d-separation: From theorems to algorithms. In *Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence (UAI'89)*, pages 139–148, New York, N. Y., 1989. Elsevier Science Publishing Company, Inc.

[72] M. Goldszmidt. Belief-based irrelevance and networks: Toward faster algorithms for prediction. In *Working notes of the AAAI-1994 Fall Symposium Series: Relevance*, pages 89–94. New Orleans, 1994.

[73] M. C. Golumbic. Triabgulated graphs. In *Algorithms Graph Theory and Perfect Graphs*, pages 98–100. Academic press, New York University, 1980.

[74] A. Grove. Two modelings for theory change. *Journal of Philosophical Logic*, pages 157–170, 1988.

[75] J. Halpern. Defining relative likelihood in partially-ordered preferential structure. *Journal of Artificial Intelligence Research*, 7:1–24, 1997.

[76] P. E. Hart and J. M. Grapham. Query-free information retreival. *IEEE Expert: Intelligent Systems and Their Applications*, 12:32–37, 1997.

[77] D. Heckerman. A tractable inference algorithm for diagnosing multiple diseases. In *Proceedings of Fifth Workshop on Uncertainty in Artificial Intelligence (UAI'89)*, pages 163–171, New York, N. Y., 1989. Elsevier Science Publishing Company, Inc.

[78] E. Hisdal. Conditional possibilities independence and non interaction. *Fuzzy Sets and Systems*, 1:283–297, 1978.

[79] E. Horvitz. Display of information for time-critical decision making. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 296–305, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[80] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 256–265, San Francisco, CA, 1998. Morgan Kaufmann Publishers.

[81] R. A. Howard and J. E. Matheson. Influence diagrams. *The Principles and Applications of Decision Analysis*, 2:720–761, 1984.

[82] C. Huangn and A. Darwiche. Inference in beleif networks: a procedural guide. *International Journal of Approximate Reasoning*, 11:1–158, 1994.

[83] T.S. Jaakkola and M.I. Jordan. Variational probabilistic inference and the qmr dt network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

[84] F. V. Jensen. *Introduction to Bayesien networks.* UCL Press, University college, London, 1996.

[85] F. V. Jensen and F. Jensen. Optimal junction trees. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI'94)*, pages 361–366, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[86] A. Kaufmann and M. Gupta. *Introduction to fuzzy arithmetic.* Theory and Publications, New York, 1985.

[87] J. H. Kim and J. Pearl. A computational model for causal and diagnostic reasoning in inference systems. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'83)*, pages 190–193, Karlsruhe (Germany), 1983.

[88] U. Kjaerulff. Triangulation of graphs - algorithms giving small total state space. In *Technical report R-90-09*. Department of mathematical and computer science, Aalborg University (Denmark), 1990.

[89] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Foundations of measurement*. Academic Press, New York and London, 1971.

[90] G. Lakemayer. Relevance from an epistemic perspective. *Artificial Intelligence*, 97:137–167, 1997.

[91] J. Lang. Possibilistic logic: complexity and algorithms. In D. Gabbay and P. Smets, editors, *Handbook on Defeasible Reasoning and Uncertainty Management Systems*, volume 5. Jürg Kohlas and Serafin Moral, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.

[92] J. Lang and P. Marquis. Complexity results for independence and definability in propositional logic. In *Proceedings of Principles of Knowledge Representation and Reasoning (KR'98)*, pages 356–367, Trento (Italie), 1998.

[93] S. L. Lauritzen. *Graphical Models*. Oxford university press, 1996.

[94] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 50:157–224, 1988.

[95] D. Lehmann. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence*, 15:61–82, 1995.

[96] D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55:1–60, 1992.

[97] D.L. Lewis. *Counterfactuals*. Basil Blackwell Ltd, Oxford, 1973.

[98] H. Moulin. *Axioms for cooperative decision-making*. Cambridge University Press, 1988.

[99] K. P. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics: Proceedings of Interface*, 33, 2001.

[100] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99)*, pages 467–475, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[101] O. Papini. Iterated revision operations stemming from the history of an agent's observations. In *Frontiers of Belief Revision*, Applied Logic Series. Kluwer ed., 2001.

[102] J. Pearl. Fusion propagation and structuring in belief networks. *Artificial Intelligence*, 29:241–288, 1986.

[103] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmman, San Francisco (California), 1988.

[104] J. Pearl and A. Paz. Graphoids: A graph-based logic for reasoning about relevance relations. In *Technical report 850038 (R-53-L)*. Cognitive systems laboratory, University of California, 1995.

[105] M. A. Poet and R. D. Shachter. Fusion and propagation with multiple observations in belief networks. *Artificial Intelligence*, 48:299–318, 1991.

[106] H. Raiffa. *Decision analysis*. Addison-Welsley Publishing Company, Toronto, 1968.

[107] R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 34:871–882, 1986.

[108] R. D. Shachter. A graph-based inference method for conditional independence. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence (UAI'91)*, pages 353–360, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[109] R. D. Shachter and M. A. Poet. Decision making using probabilistic inference methods. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence (UAI'92)*, pages 276–283, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[110] G. Shafer. *Mathematical theory of evidence*. Princston University Press, 1976.

[111] P. P. Shenoy. A fusion algorithm for solving bayesian decision problems. In *Proceedings of the Seventh Conference Uncertainty in Artificial Intelligence (UAI'91)*, pages 361–369, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[112] P. P. Shenoy. Using possibility theory in expert systems. *Fuzzy Sets and Systems*, 52:129–142, 1992.

[113] P. P. Shenoy. Valuation-based systems for bayesian decision analysis. *Operations Research*, 40:463–484, 1992.

[114] P. P. Shenoy. A comparison of graphical technical techniques for decision analysis. *European Journal of Operational Research*, 78:1–21, 1994.

[115] P. P. Shenoy. Conditional independence in valuation-based systems. *International Journal of Approximate Reasoning*, 10:203–234, 1994.

[116] P. P. Shenoy and G. Shafer. Axioms for probability and belief function propagation. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (UAI'90)*, pages 169–198, North Holland, 1990. Elsevier Science Publishers.

[117] P. Smets. Belief functions. In P. Smets, A. Mandani, and H. Prade D. Dubois, editors, *Non Standard Logics for Automated Reasoning*, pages 253–286. Academic Press, London, 1988.

[118] P. Smets and R. Kennes. The transferable belif model. *Artificial Intelligence*, 66:191–234, 1990.

[119] W. Spohn. A general non-probabilistic theory of inductive reasoning. In *Proceedings of the Fourth Conference on Uncertainty in Artificial Intelligence (UAI'88)*, pages 149–158, New York, N. Y., 1988. Elsevier Science Publishing Company, Inc.

[120] W. Spohn. Ordinal conditional functions: a dynamic theory of epistemic states causation in decision. In W. Harper and B. Skyrms, editors, *Belief Changes and Statistics*, pages 105–134. 1988.

[121] M. Studený. Formal properties of conditional independence in different calculi of artificial intelligence. In *Proceedings of Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 341–348, 1993.

[122] M. Studený. Semigraphoids and structures of probabilistic conditional independence. *Annals of Mathematics and Artificial Intelligence*, 21:71–98, 1997.

[123] M. Taniguchi, M. Haft, J. Hollmen, and V. Tresp. Fraud detection in communications networks using neural and probabilistic methods. *Siemens AG, Corporate Technology.*

[124] J. Vejnarová. Conditional independence relations in possibility theory. In *Proceedings of 1st International Symposium on Imprecise Probabilities and Their Applications (ISIPTA'99)*, Ghent (Belgique), 1999.

[125] T. Verma. Causal networks: semantics and expressiveness. In *Proceedings of the Fourth Conference on Uncertainty in Artificial Intelligence (UAI'88)*, pages 69–76, New York, N. Y., 1988. Elsevier Science Publishing Company, Inc.

[126] MA Williams. Iterated theory base change: A computational model. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1541–1547, 1995.

[127] S.K.M. Wong, C. J. Butz, and Y. Xiang. A method for implementing a probabilistic model as a relational database. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 556–564, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[128] L. A. Zadeh. Fuzzy sets. *Information and control*, 8:338–353, 1965.

[129] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information science*, 9:43–80, 1975.

[130] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[131] A. Zapico. Axiomatic foundations for qualitative/ordinal decisions with partial preferences. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 132–139, Stockholm (Sweden), 1999.

[132] N. L. Zhang, R. Qi, and D. Poole. Incremental computation of the value of perfect information in stepwise-decomposable influence diagrams. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'93)*, pages 400–407, San Francisco, CA, 1993. Morgan Kaufmann Publishers.