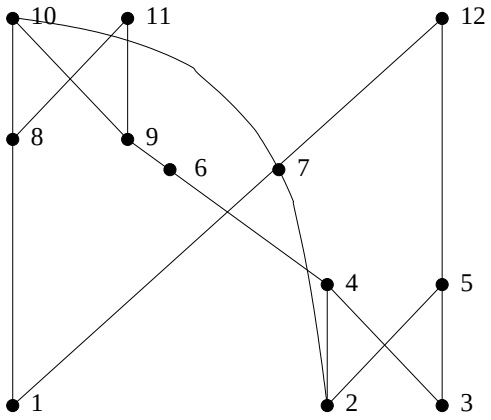
Efficient computation of rank probabilities in posets

Efficiënt berekenen van rangschikkingsprobabiliteiten in posets

Karel De Loof



Promotoren:
Prof. dr. B. De Baets
Prof. dr. H. De Meyer

Proefschrift ingediend tot het behalen van de graad van
Doctor in de Wetenschappen: Informatica

Vakgroep Toegepaste Wiskunde en Informatica
Faculteit Wetenschappen
Academiejaar 2009-2010

UNIVERSITEIT GENT

Voor Heidi en mijn ouders.

Acknowledgements

First I would like to thank my two supervisors Prof. Hans De Meyer and Prof. Bernard De Baets. During the last five years both have really given me all the support and guidance I could possibly wish for! Bernard, full of ambition and enthusiasm, was always ready with ideas and suggestions. Hans, with his catching enthusiasm and creativity, had his office door wide open for me at literally any time of the day, for discussing new ideas or just for a talk. Thank you Hans and Bernard!

I would also like to thank Prof. Gunnar Brinkmann who was always eager to do suggestions and who offered me the opportunity to use his software. I learned a lot from assisting in his courses on algorithms and data structures and he gave me the opportunity to take part in some of his research [17]. In this context I also want to thank Prof. Veerle Fack and Prof. Peter Dawyndt, whose didactic talents and refreshing views have been an inspiration. I am also greatly indebted to Dr. Rainer Brüggemann. With his characteristic enthusiasm, he introduced me to a community of mainly environmental chemists using posets. We exchanged many ideas during his visits in Ghent and the two workshops in Warsaw, which led to common publications. He also gave me the opportunity to use his software. I am grateful to Prof. Frank Ruskey for sending me his software that I could use to verify our results and to Prof. George Steiner for helping me understanding some technicalities in his paper.

A great word of thanks to all my colleagues for their friendship. In particular Davy, Stijn (whom I share or shared the office and assisted in courses with), Bert, Jan, Stéphanie (whom I assisted in courses with), Adriaan, Ester, Guy, Hilde, Koen and Michael.

Last but certainly not least, I just do not know how to thank my parents and Heidi. Without their unlimited support and love, of course, none of this would have been possible.

Karel De Loof
October 2009

Contents

Acknowledgements iii

1 Introduction **1**

2 Foundations **7**

2.1 A formal introduction to posets and ranking 7
2.2 The lattice of ideals representation of a poset 14
2.3 Constructing the lattice of ideals 17
2.3.1 A naive construction approach 17
2.3.2 The approach of Habib *et al.* based on the tree of ideals 18

3 Random generation of linear and weak order extensions **25**

3.1 Introduction 25
3.1.1 Supervised learning 25
3.1.2 Monotone classification 26
3.1.3 Training monotone classification algorithms 27
3.2 Generating rankings uniformly at random 28
3.2.1 Sampling by enumeration 28

3.2.2 Using the lattice of ideals representation 34
3.2.3 Approximate sampling 39
3.3 Generating monotone data sets 42
3.3.1 Known algorithms 43
3.3.2 Using the lattice of ideals representation 45
3.3.3 Approximate sampling 54

4 Ranking the elements of a poset 59

4.1 Introduction 59
4.2 Rank probabilities 61
4.2.1 Exact computation 62
4.2.2 Approximate computation 64
4.3 Mutual rank probabilities 67
4.3.1 Exact computation 67
4.3.2 Approximate computation 69
4.3.3 Relation between the average ranks and mutual rank probabilities 70
4.4 Real-world application 72

5 Transitivity of mutual rank probabilities 83

5.1 Introduction 83
5.1.1 Reciprocal relations 83
5.1.2 Aggregation operators 84
5.1.3 Transitivity 86
5.2 Cycle-transitivity 89
5.2.1 $T_{\mathbf{P}}$-transitivity for reciprocal relations 90
5.2.2 Definition 91

5.3 A hierarchy of types of transitivity for reciprocal relations . . . 94
5.3.1 Constant upper bound functions 94
5.3.2 Linear upper bound functions 95
5.3.3 Quadratic upper bound functions 96
5.3.4 Cubic upper bound functions 97
5.3.5 A taxonomy of types of transitivity 97
5.4 Mutual rank probability relation 99
5.5 Situating δ^*-transitivity and mutual rank transitivity in the cycle-transitivity framework 102

6 Linear extension majority cycles **111**

6.1 Counting posets with linear extension majority cycles 111
6.1.1 The algorithm 113
6.1.2 Results 116
6.2 Minimum cutting levels 120
6.2.1 Theoretical upper bounds 122
6.2.2 Experimental results 126
6.2.3 A lower bound for δ_4 128

A Posets requiring minimum cutting levels δ_m^n **141**

Conclusion **155**

Nederlandstalige samenvatting **157**

Bibliography **165**

1 Introduction

Comparing objects and inducing a ranking of them on the basis of some properties is a very common task:

- The human resource manager ranking several candidates on the basis of criteria such as experience, education, references, etc.
- The environmental expert measuring concentrations of pollutants such as lead, cadmium, zinc, ... in the herb layer of different regions in order to find out which regions are most heavily polluted.
- The bond investor trying to pick out the most creditworthy businesses using long-term debt ratings of several rating agencies.

In each example setting we have some *objects* (candidates, regions, businesses) and some *properties* (years of experience, concentration of lead in the herb layer, long-term debt rating of agency a_2, ...). For the properties of each object, a *value* is obtained after careful measurement. Remark that in reality an object is much more than just some set of properties with corresponding values. We will, however, confine ourselves to a small set of measurable properties which we think of as relevant in our context.

Let us now take a closer look at the creditworthiness example and assume our investor has five businesses (business b_1 to b_5) under consideration and has access to the publications of three rating agencies (agency a_1, a_2 and

a_3). Of course, each rating agency has its own conventions and thus uses its own rating scale:

- agency a_1 issues the ratings Aaa, Aa1, Aa2, Aa3, A1, A2, A3, Baa1, Baa2, Baa3, Ba1, Ba2, Ba3, B1, B2, B3, Caa1, Caa2, Caa3, Ca and *,
- agency a_2 issues the ratings AAA, AA+, AA, AA-, A+, A, A-, BBB+, BBB, BBB-, BB+, BB, BB-, B+, B, B-, CCC+, CCC, CCC-, CC and D,
- agency a_3 issues the ratings AAA, AA+, AA, AA-, A+, A, A-, BBB+, BBB, BBB-, BB+, BB, BB-, B+, B, B-, CCC, DDD, DD and D.

The ratings are, for each agency, listed in decreasing order, meaning the triple-A ratings are the highest ratings (highest grade), while * or D are the lowest ratings (default on debt).

Our investor consults the rating tables of each agency and summarizes her findings in Table 1.1.

	agency a_1	agency a_2	agency a_3
business b_1	Baa1	BBB+	B-
business b_2	Baa2	BBB+	AA
business b_3	Aaa	AA+	A+
business b_4	Aa1	AAA	AAA
business b_5	Aaa	AAA	AA-

Table 1.1: Ratings for each business.

We will assume that each rating agency uses its own model to decide upon a rating, and that no mistakes have been made in the decision making process. Looking at the table, it is clear that — on the basis of the information we have at our disposal — the creditworthiness of business b_5 (ratings Aaa, AAA and AA-) is superior to that of business b_1 (ratings Baa1, BBB+ and B-), since each agency issued a better rating for business b_5 than for business b_1. Less clear, however, is what can be concluded between business b_5 and business b_2: agency a_1 and a_2 estimate business b_5 more creditworthy than business b_2, but the opposite is true for agency a_3.

A solution could be to decide that if two agencies (a_1 and a_2) agree, more weight should be put on their decision than on the decision of agency a_3. As the previous sentence already suggests, this assumes that somehow *weights*

are attributed to each agency — even if these weights might be equal. If we have no a priori reason to believe that e.g. the model of agency a_1 is more faithful than the model of agency a_2, or that the combination of the model of agency a_1 and a_2 is more faithful than the model of agency a_3, there is no objective basis on which we can attribute such weights. Because of this controversy, in what follows we will refrain from assigning weights to properties of objects. We will say that we simply cannot compare business b_2 and business b_5; they are *incomparable*. Informally, we can say that, since not all objects (*i.e.* businesses) are comparable, the set of objects is a partially ordered set or *poset* for short.

Theoretically, 5! or 120 different rankings of five businesses are possible. When we take into account the known relations (*i.e.* expressions of the type "the creditworthiness of business b_i is superior to the creditworthiness of business b_j"), only 9 compatible rankings remain. Each of these rankings is not in contradiction with the information we have in Table 1.1 and is thus a valid ranking of the businesses. These remaining rankings, in increasing order of creditworthiness, are shown in Table 1.2.

$$
\begin{array}{ccccccccc}
b_1 & < & b_2 & < & b_3 & < & b_4 & < & b_5 \\
b_1 & < & b_2 & < & b_3 & < & b_5 & < & b_4 \\
b_1 & < & b_2 & < & b_4 & < & b_3 & < & b_5 \\
b_1 & < & b_3 & < & b_2 & < & b_4 & < & b_5 \\
b_1 & < & b_3 & < & b_2 & < & b_5 & < & b_4 \\
b_1 & < & b_3 & < & b_5 & < & b_2 & < & b_4 \\
b_2 & < & b_1 & < & b_4 & < & b_3 & < & b_5 \\
b_2 & < & b_1 & < & b_3 & < & b_4 & < & b_5 \\
b_2 & < & b_1 & < & b_3 & < & b_5 & < & b_4
\end{array}
$$

Table 1.2: Rankings of the businesses compatible with the ratings in Table 1.1.

In Chapter 3 of this work we will have a close look at how such a compatible ranking can be generated uniformly at random, in the sense that each ranking has exactly the same likelihood of being drawn. We will introduce an algorithm that, for a limited number of elements, performs well in practice. Furthermore, we will focus on approximative algorithms which generate such rankings almost uniformly. It will turn out that the ability of sampling uniformly at random allows for the generation of random synthetic

monotone data sets, which is important for the training and comparison of supervised ranking algorithms.

Let us return to our creditworthiness example. On the basis of the set of all compatible rankings in Table 1.2 one could easily compute the fraction of rankings where business b_2 is put on the first position. This exercise can be repeated for all businesses and positions such that all so-called *rank probabilities* are obtained. If one insists on obtaining a single ranking of all businesses a possible approach yielding such an objective ranking could be to calculate the average position for each business. These *average ranks* can then be used to obtain a single ranking. It should however be mentioned that possible ties in the ranking can occur. Indeed, it is possible that two or more businesses have an identical average rank. The rank probabilities together with the average rank for each business is shown in Table 1.3.

$b_i \backslash p$	1	2	3	4	5	average
b_1	$2/3$	$1/3$	0	0	0	$4/3$
b_2	$1/3$	$1/3$	$2/9$	$1/9$	0	$19/9$
b_3	0	$1/3$	$4/9$	$2/9$	0	$26/9$
b_4	0	0	$2/9$	$1/3$	$4/9$	$38/9$
b_5	0	0	$1/9$	$1/3$	$5/9$	$40/9$

Table 1.3: The fraction of rankings where business b_i is ranked at position p.

Recall that businesses b_2 and b_5 are incomparable. If we look at all possible compatible rankings in Table 1.2, there are 8 rankings in which business b_5 is ranked higher than business b_2 and only 1 ranking where the opposite is true. Although, on the basis of the ratings, business b_5 is not ranked higher than business b_2, there seems to be a strong preference for ranking business b_5 above business b_2. The fraction of rankings where business b_i is ranked higher than business b_j for all businesses b_i and b_j shown in Table 1.4 are the so-called *mutual rank probabilities*.

These (mutual) rank probabilities are treated in detail in Chapter 4. We will explain how they can be computed and apply the algorithms on a real-world application of herb layer pollution in Baden-Württemberg in Germany to show their power.

$b_i \backslash b_j$	b_1	b_2	b_3	b_4	b_5
b_1	0	$1/3$	0	0	0
b_2	$2/3$	0	$1/3$	0	$1/9$
b_3	1	$2/3$	0	$2/9$	0
b_4	1	1	$7/9$	0	$4/9$
b_5	1	$8/9$	1	$5/9$	0

Table 1.4: The fraction of rankings where business b_i is ranked higher than business b_j.

We can say that, the closer the mutual rank probability is to 1, the stronger is the preference, while the closer the mutual rank probability to $1/2$, the more indifference exists. As soon as the mutual rank probability that a business b_i is ranked higher than a business b_j is strictly larger than $1/2$, there is some degree of preference of business b_i over business b_j. What seems counterintuitive however is that, for a sufficiently large set of businesses, the following statements for businesses b_i, b_j and b_k can be valid at the same time:

business b_i is preferred to business b_j
business b_j is preferred to business b_k
business b_k is preferred to business b_i

Such cycles are called *linear extension majority cycles*. This cyclic behavior and closely related the transitivity of the mutual rank probabilities are discussed in Chapter 5 and 6 of this work.

2 Foundations

2.1 A formal introduction to posets and ranking

In this section we will give a formal introduction to posets and ranking. We will present the necessary definitions of the concepts that will appear in this work and illustrate them on small examples.

Definition 2.1.1. A binary relation $\leq_P$ on a set P is called a(n) *(partial) order relation* if it is

(i) reflexive: $x \leq_P x,\ \forall x \in P$
(ii) antisymmetric $x \leq_P y$ and $y \leq_P x$ imply $x =_P y,\ \forall x, y \in P$
(iii) transitive: $x \leq_P y$ and $y \leq_P z$ imply $x \leq_P z,\ \forall x, y, z \in P$

◇

If $x \leq_P y$ and $x \neq y$, we write $x <_P y$. If neither $x \leq_P y$ nor $y \leq_P x$, we say that x and y are *incomparable* and write $x ||_P y$. In case $x \leq_P y$ we can also write $y \geq_P x$.

Definition 2.1.2. A couple $(P, \leq_P)$, where P is a set of objects and $\leq_P$ is an order relation on P, is called a partially ordered set or *poset* for short. ◇

If no distinction between order relations has to be made, the index P in $\leq_P$

will be omitted. Moreover, if the order relation is clear from the context, we can simply denote the poset as P.

The *size* of a poset P, denoted by $|P|$, is defined as the number of elements in the poset. A *chain* of a poset P is a subset of P in which every two elements are comparable. Similarly, an *antichain* of a poset P is a subset of P in which every two elements are incomparable. The *height* of a poset P is defined as the size of the largest chain of P and is denoted by $h(P)$. The *width* of a poset P is defined as the size of the largest antichain of P and is denoted by $w(P)$.

Definition 2.1.3. A *linear order relation* $\leq_P$ is an order relation in which every two elements are comparable ($x \leq_P y$ or $y \leq_P x$). ◇

A linear order relation is in literature sometimes also referred to as a *complete order* or *total order*. Since a chain is a poset where every two elements are comparable, the terms chain and *linearly ordered set* cover the same concept.

Definition 2.1.4. A binary relation $\leq_P$ on a set P is called a *weak order relation* on P if it is

(i) transitive: $x \leq_P y$ and $y \leq_P z$ imply $x \leq_P z$, $\forall x, y, z \in P$
(ii) complete: $x \leq_P y$ or $y \leq_P x$, $\forall x, y \in P$
(ii') reflexive: $x \leq_P x$, $\forall x \in P$

◇

Remark that condition (ii') automatically follows from condition (ii).

A weak order relation is sometimes also called a *total preorder* or a *complete preorder*. Remark that, if a weak order relation $\leq_P$ is also antisymmetric, *i.e.* $x \leq_P y$ and $y \leq_P x$ imply $x =_P y$, $\forall x, y \in P$, it is a linear order relation.

Now let O be the set of objects to be ranked. We assume that each object $x \in O$ can be described by a tuple $q(x) = (q_1(x), q_2(x), \ldots, q_m(x))$ of values for each of the m properties, which is called an *attribute vector*, where $q_i(x) \in Q_i$ for each $i \in \{1, \ldots, m\}$. Without loss of generality, we can assume that all attribute vectors are unique. Indeed, one can always choose a single representing object from each subset of objects having equal attribute vectors. Each set Q_i is equipped with a linear order relation $\leq_i$. This reflects

the fact that q_i can be considered as a true *criterion*: if $q_i(x) \leq_i q_i(y)$, then x is as most as good as y with respect to criterion q_i. We say that x is smaller than or equal to y, denoted as $x \leq_O y$ if $q_i(x) \leq_i q_i(y)$ for all $i \in \{1, \ldots, m\}$. The relation $\leq_O$ is an order relation; it is the restriction to O of the product ordering on $Q_1 \times \ldots \times Q_m$. The objects therefore form a poset $(O, \leq_O)$.

Example 2.1. To illustrate the various concepts and algorithms, we will use the creditworthiness example from the introduction as a running example. However, from now on we will denote each business b_i as ω_i. We will call our set of objects Ω, such that $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$. Furthermore, instead of using the letter designations for the ratings, we will use ordinary numbers, where 1 corresponds to the worst rating (default on debt), and the highest number to the best rating. Each object $\omega \in \Omega$ and its representation $q(\omega)$ as a 3-tuple is given in Table 2.1.

ω_i	$q(\omega_i)$
ω_1	$(14, 14, 5)$
ω_2	$(13, 14, 18)$
ω_3	$(21, 20, 16)$
ω_4	$(20, 21, 20)$
ω_5	$(21, 21, 17)$

Table 2.1: Example set Ω with their attribute vector representation.

Here each $Q_i \subseteq \mathbb{N}$ and $\leq_i$ is the usual ordering. In our example set Ω, it holds e.g. that $\omega_1 \leq_\Omega \omega_5$ and $\omega_1 \,||_\Omega\, \omega_2$. The set Ω forms together with the partial order relation $\leq_\Omega$ the poset $(\Omega, \leq_\Omega)$. The subset $\{\omega_1, \omega_3, \omega_5\}$ of Ω constitutes a chain, while the subset $\{\omega_1, \omega_2\}$ constitutes an antichain of $(\Omega, \leq_\Omega)$. The width $w(\Omega)$ of $(\Omega, \leq_\Omega)$ is 2, since one cannot find an antichain with more than 2 elements, while the height $h(\Omega)$ of $(\Omega, \leq_\Omega)$ is 3, since $\{\omega_1, \omega_3, \omega_5\}$ is the longest chain in $(\Omega, \leq_\Omega)$. •

From here on, we consider an arbitrary finite poset $(P, \leq_P)$. For elements $x, y \in P$ we say that y *covers* x, denoted as $x \prec_P y$, if $x <_P y$ and there exists no $z \in P$ such that $x <_P z <_P y$. In other words, x is smaller than y, and no third element is situated in between x and y. An element $x \in P$ that is not covered, *i.e.* for which no $y \in P$ exists such that $x \prec_P y$, is called a *maximal element*. An element $x \in P$ that does not cover any element, *i.e.* for which no $y \in P$ exists such that $y \prec_P x$, is called a *minimal element*. Note that an element that is both minimal and maximal, is incomparable to all other

elements and can therefore be regarded as an *isolated element*.

Example 2.2. In Ω, it holds that $\omega_1 \prec_\Omega \omega_3$, but not that $\omega_1 \prec_\Omega \omega_5$ since $\omega_1 <_\Omega \omega_3 <_\Omega \omega_5$. The set of minimal elements of Ω is $\{\omega_1, \omega_2\}$ and the set of maximal elements is $\{\omega_4, \omega_5\}$. •

Definition 2.1.5. The binary relation $\prec_P$, for which it holds that $(x, y) \in \prec_P$ if and only if $x <_P y$ and there exists no $z \in P$ such that $x <_P z <_P y$, is called the *covering relation* of $(P, \leq_P)$. ◇

Definition 2.1.6. A *directed graph* G is an ordered pair (V_G, E_G), where V_G is a finite set of *vertices* and E_G is a finite set of *directed edges*, *i.e.* ordered pairs (x, y) where $x, y \in V$. A directed edge $(x, y) \in E_G$ is considered to be directed from $x \in V_G$ to $y \in V_G$, where y is called the *head* and x is called the *tail* of the directed edge. ◇

Graphs can be represented in several ways. Most commonly a directed graph is depicted by drawing a point for each vertex, where two points are connected with a line if the corresponding vertices form an edge. On each line, an arrow indicates the direction of the edge.

Example 2.3. The directed graph G with $V_G = \{v_1, v_2, v_3, v_4, v_5\}$ and $E_G = \{(v_1, v_2), (v_1, v_5), (v_2, v_3), (v_3, v_4), (v_5, v_2), (v_5, v_3)\}$ is represented in Figure 2.1. •

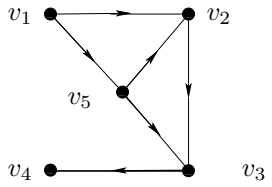


Figure 2.1: A representation of the graph G.

A poset $(P, \leq_P)$ can be conveniently represented by a directed graph C_P, called a *Hasse diagram* or *covering graph*, displaying the covering relation $\prec_P$. The set of vertices V_{C_P} is P, and the set of directed edges E_{C_P} is $\{(x, y) \in P^2 \mid x \prec_P y\}$. In a Hasse diagram, the convention is made that directed edges $(x, y) \in E_{C_P}$ are drawn upwards from x to y instead of indicating the direction with an arrow.

The graph $\bar{C}_P$ representing the *transitive closure* of the poset with a Hasse diagram C_P can be obtained by adding all edges induced by transitivity, *i.e.*

it contains all edges (x, y) such that $x <_P y$ where $x, y \in P$. In this context, the Hasse diagram C_P represents the *transitive reduction* of the poset since all edges induced by transitivity are omitted.

Example 2.4. The Hasse diagram C_Ω of $(\Omega, \leq_\Omega)$ is shown in Figure 2.2. •

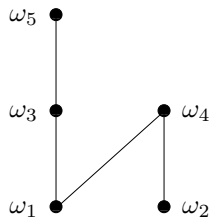


Figure 2.2: The Hasse diagram C_Ω of $(\Omega, \leq_\Omega)$.

Definition 2.1.7. A *directed walk* in a graph G from $v_0 \in V_G$ to $v_l \in V_G$ is a sequence of alternating vertices and edges $(v_0, e_1, v_1, e_2, \ldots, v_{l-1}, e_l, v_l)$, such that for each edge e_i from the sequence, v_i is the head and v_{i-1} is the tail, where $i = 1, \ldots, l$. The number of edges l in the walk is called the *length* of the walk. If in a directed walk from $v_0 \in V_G$ to $v_l \in V_G$ it holds that $v_0 = v_l$, it is called a *closed walk*. ◇

Definition 2.1.8. A *directed path* is a directed walk in which no repeating vertices occur, except possibly the first and last vertex. A directed path where the first and the last vertex are equal is called a *directed cycle*. ◇

Since only directed graphs are considered in this work, for the sake of brevity we will use the terms walk, path and cycle where we in fact mean directed walk, directed path and directed cycle. Note that the *length* of a path is defined as the number of edges in the path.

Remark that $x <_P y$ if and only if there is a path from x to y in the Hasse diagram C_P.

Definition 2.1.9. A *directed weighted graph* G is defined as a triplet comprising of a set V_G of vertices, a set E_G of edges and a *weight function* $w_G : E_G \to \mathbb{R}$. To each edge $e \in E_G$ a *weight* $w_G(e)$ is attached. ◇

Definition 2.1.10. A *tree* T is a directed graph in which any two vertices $x, y \in V_T$ are connected by exactly one path if the directions of the edges would be ignored. All edges are directed away from a specific vertex $r \in V_T$,

called the *root* of T, *i.e.* there is a unique path from r to each vertex $v \in V_T$. In a tree, a vertex is usually called a *node*. ◇

Example 2.5. The tree T with $V_T = \{v_1, v_2, v_3, v_4, v_5\}$ and $E_T = \{(v_1, v_2), (v_1, v_5), (v_2, v_3), (v_3, v_4)\}$ rooted in v_1 is depicted in Figure 2.3. •

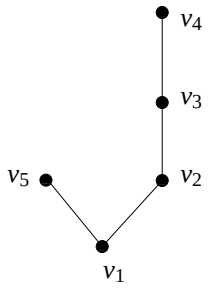


Figure 2.3: A representation of the tree T with root v_1.

Remark that since v_1 is the root of T, the directions of the edges are fixed. Therefore, if the root vertex is clear from the context, it is not necessary to indicate edge directions. The tree T is called a *spanning tree* of the graph G, since T is a tree for which $V_T = V_G$ and $E_T \subseteq E_G$.

Definition 2.1.11. An ordering $(x_1, x_2, ..., x_n)$ of the elements of a poset $(P, \leq_P)$ which is consistent with the partial order $\leq_P$, that is, such that $x_i \leq_P x_j$ implies $i \leq j$, is called a *linear extension* of P. ◇

More generally, a poset $(Q, \leq_Q)$ is called an *extension* of $(P, \leq_P)$ if $Q = P$ and if $x \leq_P y$ implies that $x \leq_Q y$. A linear extension is an extension in which every two elements are comparable. Each linear extension of P corresponds to a possible ranking of the elements of P which respects the order relation $\leq_P$. Both concepts are essentially the same; in this work we will use the term linear extension from a formal and algorithmic point of view, and use the term ranking from an application point of view. Let us denote the set of all linear extensions of a poset P as $\mathcal{E}(P)$. We denote the number of linear extensions, *i.e.* the cardinality of $\mathcal{E}(P)$, as $e(P)$.

Example 2.6. In Table 2.2 the set $\mathcal{E}(\Omega)$ of the 9 linear extensions of our example poset Ω is shown. According to the Hasse diagram of Figure 2.2, these are all rankings in which w_3 does not appear before w_1, w_5 does not appear before w_3 and w_4 does not appear before w_1 and w_2. Remark that

these linear extensions are the compatible rankings in Table 1.2 in the introduction of this work. •

$(\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$	$(\omega_1, \omega_3, \omega_5, \omega_2, \omega_4)$
$(\omega_1, \omega_2, \omega_3, \omega_5, \omega_4)$	$(\omega_2, \omega_1, \omega_4, \omega_3, \omega_5)$
$(\omega_1, \omega_2, \omega_4, \omega_3, \omega_5)$	$(\omega_2, \omega_1, \omega_3, \omega_4, \omega_5)$
$(\omega_1, \omega_3, \omega_2, \omega_4, \omega_5)$	$(\omega_2, \omega_1, \omega_3, \omega_5, \omega_4)$
$(\omega_1, \omega_3, \omega_2, \omega_5, \omega_4)$	

Table 2.2: All linear extensions of $(\Omega, \leq_\Omega)$.

In literature, several subclasses of posets are defined. In this work we will consider two such subclasses, namely interval orders and series-parallel posets.

Definition 2.1.12. A poset $(P, \leq_P)$ is an *interval order* [56] if there is no subset $\{x_1, x_2, x_3, x_4\} \subseteq P$ with $x_1 <_P x_3$ and $x_2 <_P x_4$ being the only comparabilities. ◇

It can be shown that a poset $(P, \leq_P)$ is an interval order if and only if there exists a bijection from P to a set of real intervals mapping each $x_i \in P$ to an interval $[l_i, r_i]$ such that for any $x_i, x_j \in P$ we have $x_i <_P x_j$ when $r_i <_{\mathbb{R}} l_j$.

Definition 2.1.13. A poset $(P, \leq_P)$ is a *linear sum* if there exist disjoint non-empty subsets P_1 and P_2 of P such that $P = P_1 \cup P_2$, and $x <_P y$ if $x, y \in P_1$ and $x <_{P_1} y$, or $x, y \in P_2$ and $x <_{P_2} y$, or $x \in P_1$ and $y \in P_2$. ◇

Definition 2.1.14. A poset $(P, \leq_P)$ is a *disjoint sum* if there exist disjoint non-empty subsets P_1 and P_2 of P such that $P = P_1 \cup P_2$, and $x <_P y$ if either $x, y \in P_1$ and $x <_{P_1} y$ or $x, y \in P_2$ and $x <_{P_2} y$. ◇

Definition 2.1.15. A poset $(P, \leq_P)$ is called *series-parallel* if there is no subset $\{x_1, x_2, x_3, x_4\} \subseteq P$ for which $x_1 <_P x_3$, $x_1 <_P x_4$ and $x_2 < x_4$ are the only comparabilities. ◇

It can be shown that a poset $(P, \leq_P)$ is series-parallel if it can be recursively constructed by applying the operations of linear and disjoint sum, starting with a single element.

Example 2.7. The poset $(\Omega, \leq_\Omega)$, whose Hasse diagram is shown in Figure 2.2, is not an interval order since the only comparabilities in the subset $\{\omega_2, \omega_3, \omega_4, \omega_5\} \subseteq \Omega$ are $\omega_3 <_\Omega \omega_5$ and $\omega_2 <_\Omega \omega_4$, neither is it a series-parallel poset since the subset $\{\omega_1, \omega_2, \omega_3, \omega_4\} \subseteq \Omega$ has exclusively the comparabilities $\omega_1 <_\Omega \omega_3$, $\omega_1 <_\Omega \omega_4$ and $\omega_2 <_\Omega \omega_4$. •

We refer the interested reader for a more extensive overview of poset theory to [32].

2.2 The lattice of ideals representation of a poset

It will turn out in the remainder of this work that the so-called lattice of ideals representation of a poset is playing a key role in many of the algorithms presented. In this section this lattice of ideals representation is introduced.

Definition 2.2.1. A *downset* or *ideal* of a poset $(P, \leq_P)$ is a subset $D \subseteq P$ such that $x \in D$, $y \in P$ and $y \leq_P x$ imply $y \in D$. Dually, an *upset* or *filter* of a poset $(P, \leq_P)$ is a subset $U \subseteq P$ such that $x \in U$, $y \in P$ and $x \leq_P y$ imply $y \in U$. ◇

Note that both ideals and filters are *order-convex*, in the sense that if $x \leq_P y$ belong to a given ideal or filter, then so does any z such that $x \leq_P z \leq_P y$. The ideal of P consisting of all elements smaller than or equal to $x \in P$ is denoted as $\downarrow x$; the filter of P consisting of all elements greater than or equal to $x \in P$ is denoted as $\uparrow x$. Let us denote the set of all ideals of a poset $(P, \leq_P)$ as $\mathcal{I}(P)$ and its cardinality as $i(P)$.

Example 2.8. In $(\Omega, \leq_\Omega)$, the subsets $\{\omega_1, \omega_2, \omega_3\}$ and $\{\omega_1, \omega_2, \omega_4\}$ are both ideals, while $\{\omega_1, \omega_3, \omega_4\}$ is not an ideal since $\omega_2 \leq_\Omega \omega_4$ and as such, ω_2 should also reside in the ideal. The set of ideals $\mathcal{I}(\Omega)$ is given in Table 2.3. •

If for every $x, y \in P$ the greatest lower bound $x \wedge y$ (read "*x meet y*") and the smallest upper bound $x \vee y$ (read "*x join y*") exist, a poset $(P, \leq_P)$ is called a *lattice*. An element $x \in P$ in a lattice is called *meet irreducible* if x is not maximal in P and $x = a \wedge b$ implies $x = a$ or $x = b$ for any $a, b \in P$. Dually, an element $x \in P$ is called *join irreducible* if x is not minimal in P and $x = a \vee b$ implies $x = a$ or $x = b$ for any $a, b \in P$. In other words, an

$\emptyset$	$\{\omega_1, \omega_2, \omega_4\}$
$\{\omega_1\}$	$\{\omega_1, \omega_3, \omega_5\}$
$\{\omega_2\}$	$\{\omega_1, \omega_2, \omega_3, \omega_4\}$
$\{\omega_1, \omega_2\}$	$\{\omega_1, \omega_2, \omega_3, \omega_5\}$
$\{\omega_1, \omega_3\}$	$\{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$
$\{\omega_1, \omega_2, \omega_3\}$	

Table 2.3: All ideals of $(\Omega, \leq_\Omega)$.

element is meet irreducible if it is covered by at most one other element and is join irreducible if it covers at most one other element. A *distributive lattice* is a lattice that satisfies distributivity: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

If we consider the set of ideals $\mathcal{I}(P)$ and equip it with the set inclusion $\subseteq$, which is a partial order relation, a new poset $(\mathcal{I}(P), \subseteq)$ is obtained. Moreover, it is more than a poset: it is a distributive lattice. Distributive lattices exhibit several interesting properties. The interested reader is referred to [32] for an extensive overview on (distributive) lattices. The distributive lattice $(\mathcal{I}(P), \subseteq)$ is called the *lattice of ideals* of $(P, \leq_P)$. Note that the lattice of ideals plays an important role in formal concept analysis [63].

Example 2.9. The Hasse diagram L_Ω of $(\mathcal{I}(\Omega), \subseteq)$ is shown in Figure 2.4. •

The lattice of ideals $(\mathcal{I}(P), \subseteq)$ is an alternative representation of the poset $(P, \leq_P)$. From the poset $(P, \leq_P)$ the lattice of ideals $(\mathcal{I}(P), \subseteq)$ can be constructed, neither is it difficult to reconstruct $(P, \leq_P)$ from $(\mathcal{I}(P), \subseteq)$. In Section 2.3 it will be shown how the lattice of ideals of a poset can be constructed efficiently.

Note that the Hasse diagram of the lattice of ideals can be considered as a directed graph L_P with a unique minimal element, called the *source* and denoted as $v_\perp$, and a unique maximal element, called the *sink* and denoted as $v^\top$. We will denote by I_f the $V_{L_P} \rightarrow \mathcal{I}(P)$ function that maps a vertex $v \in V_{L_P}$ to the corresponding ideal in $(\mathcal{I}(P), \subseteq)$.

Remark 2.2.2. An important property of the Hasse diagram L_P of the lattice of ideals $(\mathcal{I}(P), \subseteq)$ should be noted: there exists a one-to-one correspondence between the linear extensions of a poset $(P, \leq_P)$ and the maximum-length paths from $v_\perp$ to $v^\top$ in L_P [2, 10]. First, we note that each edge in

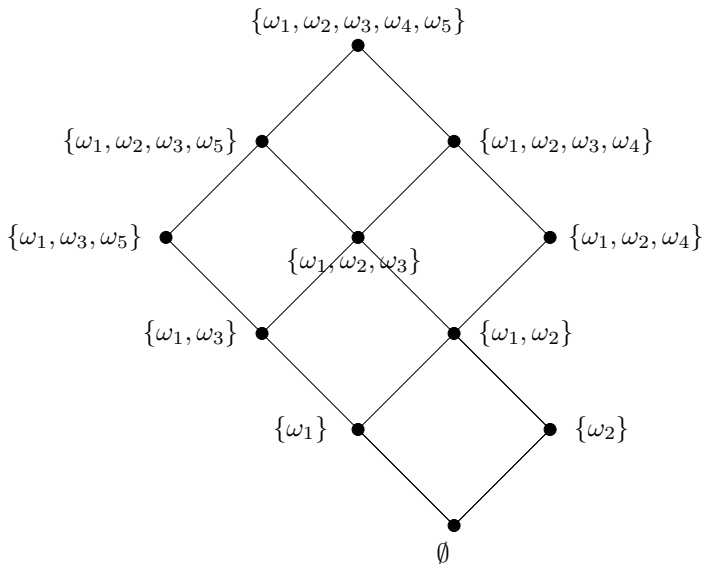


Figure 2.4: The Hasse diagram L_Ω of $(\mathcal{I}(\Omega), \subseteq)$.

L_P connecting a vertex v_1 for which $I_f(v_1) = I_1 \in \mathcal{I}(P)$ and a vertex v_2 for which $I_f(v_2) = I_2 \in \mathcal{I}(P)$, with $I_1 \subset I_2$, can be labeled with the unique element of $I_2 \setminus I_1$. If we follow a path from $v_\perp$ to $v^\top$ in L_P, the order in which each label is encountered induces a linear extension of $(P, \leq_P)$. This one-to-one correspondence will be used extensively in the remainder of this work and is formalized in Lemma 2.2.3.

Lemma 2.2.3 (Bonnet and Pouzet [10])**.** *Let $(P, \leq_P)$ be a poset, then the set of linear extensions $(L, \leq_L)$ of P is in one-to-one correspondence with the set of maximum-length paths of the Hasse diagram L_P of $(\mathcal{I}(P), \subseteq)$. More precisely, if $(L, \leq_L)$ is a linear extension of P, then the Hasse diagram L_L of $(\mathcal{I}(L), \subseteq)$ is a maximum-length path of L_P. On the other hand, for each maximum-length path corresponding to a chain $(C, \subseteq)$ of $(\mathcal{I}(P), \subseteq)$, there exists a unique linear extension $(L, \leq_L)$ of $(P, \leq_P)$ such that $(C, \subseteq) = (\mathcal{I}(L), \subseteq)$.*

Example 2.10. The highlighted path of L_Ω in Figure 2.5 corresponds to the linear extension $(\omega_1, \omega_3, \omega_2, \omega_5, \omega_4)$. Note that the edges are labeled as described in Remark 2.2.2. •

Definition 2.2.4. Two graphs G_1 and G_2 are called *isomorphic* if there exists a bijection $\phi : V_{G_1} \to V_{G_2}$ such that for all $v_1, v_2 \in V_{G_1}$ it holds that (v_1, v_2)

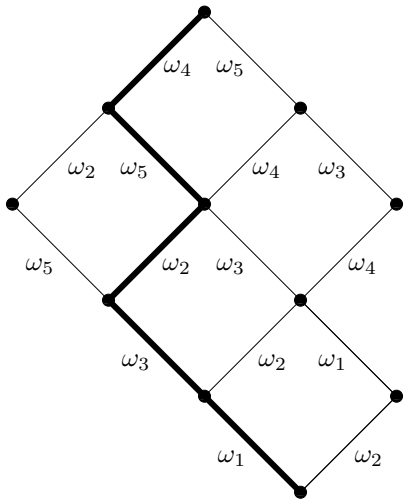


Figure 2.5: The Hasse diagram L_Ω of $(\mathcal{I}(\Omega), \subseteq)$ with labeled edges and a highlighted maximum-length path.

is an edge in G_1 if and only if $(\phi(v_1), \phi(v_2))$ is an edge in G_2. ◇

Finally, it is important to mention that for any finite distributive lattice the subset of join- (or meet-)irreducible elements is a poset such that the Hasse diagram of its lattice of ideals is isomorphic to the Hasse diagram of the distributive lattice [9].

Example 2.11. The poset of join- (or meet-)irreducible elements of L_Ω in Figure 2.4 is the poset $(\Omega, \leq_\Omega)$ depicted in Figure 2.2. The other way around, L_Ω is the Hasse diagram of the lattice of ideals of $(\Omega, \leq_\Omega)$. •

2.3 Constructing the lattice of ideals

2.3.1 A naive construction approach

We start with a naive but intuitive algorithm to build the Hasse diagram L_P of the lattice of ideals of a given poset $(P, \leq_P)$, before providing a more

complex, yet more efficient one. The algorithm gradually constructs the lattice of ideals, layer per layer, from the empty ideal $\emptyset$ to the maximum ideal P.

We start with an empty graph (V, E), where V is the set of vertices and $E \subseteq V^2$ the set of edges and start with an empty list $L = \emptyset$ of sets. Initially, V only contains the vertex $v_\emptyset$ corresponding to the empty ideal. For each minimal element $x \in P$, we add $\{x\}$ to L, add a vertex $v_{\{x\}}$ (corresponding to the ideal $\{x\}$) to V and add an edge $(v_\emptyset, v_{\{x\}})$ to E. Next, we perform the following steps until $L = \emptyset$. We remove the first set $I \in L$. For each minimal element $x \in P \setminus I$ we do the following: if the ideal $I' = I \cup \{x\}$ is not in L, then we add I' to the end of L and add $v_{I'}$ (corresponding to the ideal I') to V, and add $(v_I, v_{I'})$ to E; if $I' \in L$, we just add $(v_I, v_{I'})$ to E.

At the end of this algorithm, the graph (V, E) is the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$. While this algorithm is easy to follow, it is not very efficient. Especially the repeated computation of the minimal elements of a subposet of P, and the check whether $I' \in L$ is time-consuming.

2.3.2 The approach of Habib *et al.* based on the tree of ideals

A more efficient approach [70] uses a so-called tree of ideals [71] as an intermediate representation to construct the lattice of ideals. A *tree of ideals* T_P of a poset $(P, \leq_P)$ is a spanning tree of the Hasse diagram L_P of the lattice of ideals $(\mathcal{I}(P), \subseteq)$ that is rooted in $v^\top$. Since T_P is a spanning tree, it contains all vertices of L_P and has a unique path from the root $v^\top$ of T_P to each other vertex in L_P.

Assuming that the labeling of the edges of L_P described in Remark 2.2.2 is available, starting from the Hasse diagram L_P a tree of ideals can be obtained as follows. Initially, an arbitrary linear extension σ of the poset $(P, \leq_P)$ is chosen. Once this choice has been made, the tree of ideals representation is uniquely determined by the set of all paths of L_P starting from $v^\top$ whose labeling, *i.e.* edge labels ordered according to their occurrence in each path, respects the reverse order of σ.

Example 2.12. The tree of ideals T_Ω of $(\Omega, \leq_\Omega)$ corresponding to the initial linear extension $(\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ is shown in Figure 2.6. Remark that

in each path from the root to a vertex the reverse order of the chosen linear extension is respected. Moreover, there is exactly one path where each of the elements of Ω is present. •

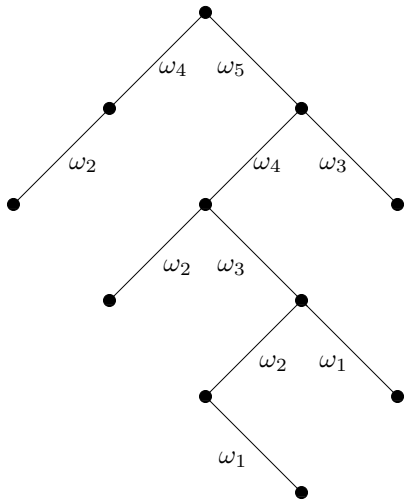


Figure 2.6: The tree of ideals representation T_Ω of L_Ω corresponding to the initial linear extension $(\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$.

Without loss of generality, in the following we will assume that for a poset $(P, \leq_P)$ its set of elements is given by $P = \{1, 2, \ldots, n\}$ and that $\sigma = (1, 2, \ldots, n)$ is a linear extension of $(P, \leq_P)$. We will furthermore define $B_P(i) = \{j \in P \mid j \,||_P\, i \ \wedge\ j <_{\mathbb{N}} i\}$, where $\mathbb{N}$ is the natural ordering of integers, or $B(i)$ for short if the poset is clear from the context. The set of *predecessors* of an element $i \in P$ will be denoted as $\mathrm{Pred}(i) = \{j \mid j <_P i\}$ and the set of *successors* of i as $\mathrm{Succ}(i) = \{j \mid i <_P j\}$. We will furthermore define the set of *immediate predecessors* $\mathrm{ImPred}(i) = \{j \mid j \prec_P i\}$ and the set of *immediate successors* $\mathrm{ImSucc}(i) = \{j \mid i \prec_P j\}$.

2.3.2.1 Constructing the tree of ideals

Let us denote the tree of ideals of $(P, \leq_P)$ fixed by the linear extension σ as T_P. In this subsection, the algorithm of Habib *et al.* to incrementally construct the tree of ideals T_P from $(P, \leq_P)$ is presented. We assume that

each element $i \in P$ is added in the order of appearance in σ and denote by $P + \{i\}$ the poset obtained from P by adding an element i.

For each vertex v in T_P where $v \neq v^\top$ we define:

- $\mathrm{Parent}(v)$: the vertex immediately before v in the unique path in the tree from $v^\top$ to v;
- $\mathrm{Label}(v)$: the label attached to the vertex v.

Let us associate with each vertex $v \in V_{T_P}$ corresponding to an ideal $I \in \mathcal{I}(P)$, *i.e.* $I = I_f(v)$, where $v \neq v^\top$, a label defined by $\mathrm{Label}(v) = \min_\sigma(P \setminus I)$. In other words, we look at the minimal elements of $P \setminus I$ and take the minimal element in the linear extension σ as a label for I.

Example 2.13. Remark that the labels attached to each vertex are precisely the labels attached to the incoming edges in T_Ω depicted in Figure 2.6. In this way, all vertices get labels except the root vertex of T_Ω. •

For each vertex v in T_P we furthermore define:

- $\mathrm{Child}(v)$: the list of children of v in T_P, *i.e.* the vertices $v_i \in T_P$ such that $(v, v_i) \in E_{T_P}$, sorted in decreasing order of their labels.

We recursively construct the tree of ideals $T_{P+\{i\}}$ from T_P. Since the ideals $I_f(v)$, for $v \in V_{T_P}$, do not contain i, the remaining ideals that should be generated are the ideals containing i and thus $\downarrow i$. If I is an ideal for which $\downarrow i \subseteq I$ then I can be partitioned into the disjoint union of $\downarrow i$ and the set $I \setminus \downarrow i$ of remaining elements such that I is an ideal of $P + \{i\}$ if and only if $I \setminus \downarrow i$ is an ideal of $B(i)$. The remaining ideals to be generated are thus the ideals of $B(i)$. From the above argument, it can easily be seen that the tree of ideals corresponding to $B(i)$ is isomorphic to a subtree of T_P rooted at $v^\top$. Indeed, each ideal J of $B(i)$ corresponds to another ideal $J \cup \mathrm{Pred}(i)$ of P.

In order to connect the tree of ideals corresponding to $B(i)$ and the tree of ideals T_P, a glueing operation $\bowtie^i$ is defined.

Definition 2.3.1. The *glueing operation* $T \bowtie^i T'$ of two trees T and T' is defined as follows:

- connect the root of T as the first child of the root of T' (as to guarantee that children are sorted according to the inverse order of σ)

- label the root r of T as i, *i.e.* $\text{Label}(r) = i$

$\diamond$

It holds that $T_{P+\{i\}} = T_P \bowtie^i T_{B(i)}$ where i is a maximal element of $P + \{i\}$. Therefore, to construct T_P one should copy and glue subtrees recursively, starting with one vertex corresponding to the empty ideal $\emptyset$.

Algorithm 2.1 implements this idea. In line 4 of the function Left the left-most subtree r containing all ideals that do not include i is created. In lines 6-8 this left tree r and the right tree *root* (containing the ideals including i) are glued together as described in Definition 2.3.1. Procedure Right constructs the remainder of the tree where all ideals contain i. In order to know whether we have an ideal $J \cup \text{Pred}(i)$, where J is an ideal of $B(i)$, we check in line 1 that there is no $s \in V_{T_P}$ on the path from $v^\top$ to the vertex v corresponding to this ideal, such that $\text{Label}(s) \in \text{ImPred}(i)$. Indeed, if such a vertex $s \in V_{T_P}$ would exist, the ideal would not contain $\text{Label}(s) \in \text{ImPred}(i)$, and thus not contain $\text{Pred}(i)$.

Theorem 2.3.2 (Habib *et al.* [70])**.** *Algorithm 2.1 constructs a tree of ideals $T_P = (V_{T_P}, E_{T_P})$ from the poset $(P, \leq_P)$ in time $\mathcal{O}(\Delta(P) \cdot |V_{T_P}|)$, where $\Delta(P)$ is defined as* $\max_{x \in P} |\text{ImPred}(x)|$.

2.3.2.2 Constructing the lattice of ideals from the tree of ideals

Let T_P be a tree of ideals of $(P, \leq_P)$ and $v \in V_{T_P}$. In order to represent the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$, we extend the data structure representing the tree of ideals by defining:

- $\text{ImPred}(v)$: all vertices $v' \in V_{T_P}$ for which $(v', v) \in E_{L_P}$.

It can be shown that $\text{ImPred}(v)$ equals the union of the set $\text{Child}(v)$ and the set

$$\{\text{Child}(v', k) \mid v' \in \text{ImPred}(\text{Parent}(v)) \text{ and } \text{Label}(v') > k\}$$

where $\text{Child}(v', k)$ denotes the child of v' labeled with k and $k = \text{Label}(v)$.

This observation leads to Algorithm 2.2, where the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ is constructed on the basis of a tree of ideals T_P.

Algorithm 2.1 Constructing a tree of ideals T_P of a poset $(P, \leq_P)$

compute a linear extension of $(P, \leq_P)$: $\sigma \leftarrow (1, 2, \ldots, n)$
return Left(n)

function Left(Integer i)

create a new vertex *root*
if $i = 0$ **then**
    return *root*
$r \leftarrow \text{Left}(i-1)$
Right($i, r,$ *root*)
Parent(r) $\leftarrow$ *root*
add r as the left child of *root*
Label(r) $\leftarrow i$
return *root*

procedure Right(Integer i, Vertex r, Vertex *root*)

for each child s of r such that Label(s) $\notin$ ImPred(i) **do**
    create a copy t of s
    Parent(t) $\leftarrow$ *root*
    add t as the left child of *root*
    Label(t) $\leftarrow$ Label(s)
    Right(i, s, t)

In line 3 the vertices $v \in V_{T_P}$ are sorted according to their label $\text{Label}(v)$ and lines 4-5 iterate over the vertices $v \in V_{T_P}$ in decreasing order of their label. In this way, it is guaranteed that $\text{Label}(v') > k = \text{Label}(v)$ on lines 7-11. In line 14 the vertex v is deleted from the list of children of its parent, as to guarantee that children with greater labels are deleted before children with smaller labels. Therefore, in line 9, vertex v^* with $\text{Label}(v^*) = k$ will be the first child of v'.

Algorithm 2.2 Construction of the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ based on a tree of ideals T_P

procedure BuildLattice(Ideal tree T_P)

1: $\text{ImPred}(v^\top) \leftarrow \text{Child}(v^\top)$
2: **for** each $k \in \{1, 2, \ldots, n\}$ **do**
3: compute the list $E_k = \{v \in T_P \mid \text{Label}(v) = k\}$
4: **for** $k = n, n-1, \ldots, 1$ **do**
5: **for** each $v \in E_k$ **do**
6: $\text{ImPred}(v) \leftarrow \emptyset$
7: $v' \leftarrow$ first element in $\text{ImPred}(\text{Parent}(v))$
8: **while** $v' \neq v$ **do**
9: $v^* \leftarrow$ first child of v'
10: $\text{ImPred}(v) \leftarrow \text{ImPred}(v) \cup \{v^*\}$
11: $v' \leftarrow$ next element in $\text{ImPred}(\text{Parent}(v))$
12: $\text{ImPred}(v) \leftarrow \text{ImPred}(v) \cup \text{Child}(v)$
13: **for** each $v \in E_k$ **do**
14: delete v from the list of children of its parent

Theorem 2.3.3 (Habib *et al.* [70])**.** *Algorithm 2.2 constructs from a tree of ideals T_P of a poset $(P, \leq_P)$ the Hasse diagram L_P of the lattice of ideals in time $\mathcal{O}(|V_{L_P}| + |E_{L_P}|)$.*

3 Random generation of linear and weak order extensions

3.1 Introduction

3.1.1 Supervised learning

Machine learning is a field concerned with the analysis and development of algorithms that aim to learn from data by combining aspects from artificial intelligence, data mining and statistics. The focus in machine learning is on a predictive analysis of the data, meaning that a model capable of learning trends from the data is built. Over time, the predictive power of the algorithms is expected to increase as more training data is provided. This approach can be useful in a large number of applications, some of which are:

- Weather forecasting: predicting several parameters such as pressure, temperature, rainfall.
- Speech and handwriting recognition: transforming human speech and handwriting into plain text.
- Fraud detection: warning when potentially fraudulent money transfers are executed.
- Bioinformatics: localizing genes in DNA sequences corresponding to

some characteristics.

- Game playing: making a good move in a chess play against a human.
- Spam filtering: classifying unsolicited e-mails in a separate folder of a mailbox.

In all of the examples, the aim is to represent the available data in a statistical way such that the resulting model gives satisfactory predictions. Depending on the structure of the incoming data, one subdivides machine learning in two categories: supervised and unsupervised learning.

In *supervised learning*, the incoming data contains feedback information for which we already know the desired output. Referring to the examples of machine learning given above, this feedback information could be:

- Weather forecasting: the complete weather survey of the last 30 years.
- Speech and handwriting recognition: several fragments of different people's speech and handwriting together with the plain text translation.
- Fraud detection: a list of known fraudulent transfers in the last 10 years.
- Bioinformatics: a list of genes detected in previous research.
- Game playing: a database of the sequences of moves in several chess games.
- Spam filtering: a huge amount of known unsolicited and legitimate e-mails with their classification.

In *unsupervised learning* there is no such kind of feedback information. A model must therefore be built solely on the basis of data without the desired output.

3.1.2 Monotone classification

In a *classification problem*, to each element of a set of objects a label must be assigned [91, 120].

Let us denote the set of labels as L, and the set of objects, represented by

their attribute vectors $q(x) = (q_1(x), q_2(x), \ldots, q_m(x))$, where $q_i(x) \in Q_i$ for each $i \in \{1, \ldots, m\}$, as O. Each set Q_i is equipped with a linear order relation $\leq_i$ reflecting the fact that q_i can be considered as a true criterion. Only the case is considered where the labels $l \in L$ are quality judgements. Therefore we will insist that the set L is equipped with a linear order relation.

Definition 3.1.1. A *classification rule* on O is defined as a mapping $d : O \to L$, where O is the set of objects and L the set of labels. ◇

Definition 3.1.2. A classification rule $d : O \to L$ is called *monotone* if $x < y \Rightarrow d(x) \leq_L d(y)$ for all $x, y \in O$. ◇

In the above multi-criteria context, where attributes are criteria and labels represent quality judgements, the classification rule $d : O \to L$ is required to be monotone. This kind of classification is referred to as *monotone classification* or *ordered sorting*.

Remark 3.1.3. Without this monotonicity constraint a situation could occur in which for two objects $x, y \in O$ it holds that $x <_O y$ but $d(x) >_L d(y)$. Consider two candidates A and B for a job. Candidate A performs better than candidate B on all criteria, but nevertheless is assigned a label which is strictly worse than the label assigned to candidate B. This would be at least considered unfair, and it shows that it is natural to add the monotonicity constraint in this multi-criteria context.

Remark 3.1.4. Ranking is a specific case of monotone classification: one where the set of labels is a set $\{1, 2, \ldots, n\}$ with $n = |O|$ and the label assigned to an object corresponds to the position of the object in the ranking.

Supervised learning algorithms that generate such monotone classification rules are often referred to as *monotone classification algorithms*, *ranking algorithms* or *ordered sorting algorithms* (see e.g. [3, 4, 5, 6, 27, 86, 87, 88]).

3.1.3 Training monotone classification algorithms

In order to properly train a monotone classification algorithm, a set of learning examples must be provided.

Definition 3.1.5. *Learning examples* are couples $(p, \alpha) \in O \times L$ consisting of an attribute vector $p \in O$ and a label $\alpha \in L$. ◇

Learning examples can be obtained from a data set. However, real-world data sets (see e.g. [117]) are often pervaded with some degree of *noise* (see e.g. [27, 86]):

- *doubt*: $(p, \alpha) \in O \times L$ and $(p, \beta) \in O \times L$, with $\alpha \neq \beta$,
- *reversed preference*: $(p, \alpha) \in O \times L$ and $(q, \beta) \in O \times L$, with $p <_O q$ and $\alpha >_L \beta$.

Especially the latter is difficult to quantify or remove (see e.g. [30, 103, 104]). Such data sets are not suitable for initial testing of newly designed classification algorithms on their capacity of modelling monotone relationships. Moreover, several monotone classification algorithms do not accept data sets containing the types of noise mentioned. Hence, both for testing and comparison purposes, there is a need for randomly generated synthetic data sets that are noise-free. Data sets that are free of the types of noise mentioned are called *monotone data sets*.

3.2 Generating rankings uniformly at random

In the artificial case where there are as many labels to be assigned as objects, *i.e.* the mapping of objects to labels is bijective, a dataset consists, next to a poset of objects and a set of labels, of a monotone mapping defined by a linear extension of $(O, \leq_O)$. Indeed, a linear extension is a ranking of the elements of O that respects the underlying order $\leq_O$, and therefore characterizes a monotone mapping from O to L.

In this section we will therefore address the problem of generating rankings uniformly at random. In solving this specific case, the foundations for generating monotone data sets are laid. Besides, the ability of generating one or more rankings uniformly at random is of importance in itself.

3.2.1 Sampling by enumeration

The most straightforward approach to sample objects, which in our case are rankings, uniformly at random is to enumerate all objects and to select a

subset of objects in a uniform way. In this subsection we will therefore focus on the problem of enumerating such rankings as well as on obtaining the number of such rankings.

3.2.1.1 Enumerating all rankings

The problem of efficiently generating all linear extensions has been studied in the papers of Knuth and Szwarcfiter [80], Varol and Rotem [116] and Kalvin and Varol [77]. Note that they use the term *topological sort* instead of linear extension. Of these algorithms, the algorithm of Varol and Rotem is surprisingly simple and fast in practice.

Let $(P, \leq_P)$ be a poset with n elements. Without loss of generality, we can assume that for a poset $(P, \leq_P)$ its set of elements is given by $P = \{1, 2, \ldots, n\}$ and that $(1, 2, \ldots, n)$ is a linear extension of $(P, \leq_P)$. Suppose a list L_{n-1} of all extensions of $P \setminus \{n\}$ is available. Now the list L_n of all extensions of $(P, \leq_P)$ can be obtained in the following way.

Let $\pi = (\pi_1, \pi_2, \ldots, \pi_{n-1}) \in L_{n-1}$, and k the maximum index for which $\pi_k \prec_P n$. Assume furthermore that there is an element π_0 less than all other elements in the poset. Since $\pi_j \,||_P\, n$ for all $j > k$, the permutations

$$(\pi_1, \ldots, \pi_k, \pi_{k+1}, \ldots, \pi_j, n, \pi_{j+1}, \ldots, \pi_{n-1})$$

where j varies from n to k, form a list of all extensions of $(P, \leq_P)$ where the elements of $P \setminus \{n\}$ occur in the order of π. When $j = k$, the original permutation π can be restored by rotating the elements of $(n, \pi_{k+1}, \ldots, \pi_{n-1})$ to the left such that we obtain $(\pi_{k+1}, \ldots, \pi_{n-1}, n)$. Subsequently the next extension in L_{n-1} is taken. This process continues until all extensions in L_{n-1} have been used. At this point, all linear extensions of $(P, \leq_P)$ will have been generated.

This algorithm is listed in Algorithm 3.1. Initially $\pi = (1, 2, \ldots, n)$, and the procedure is invoked as $\mathrm{VR}(1)$. The running time of the algorithm depends on the linear extension that is chosen to initialize the algorithm. In some cases, when a 'good' choice for the initial linear extension has been made, the algorithm needs only $\mathcal{O}(e(P))$ time to enumerate all linear extensions, while examples can be easily constructed where it needs $\mathcal{O}(n \cdot e(P))$ time.

Algorithm 3.1 The Varol-Rotem algorithm for generating all linear extensions of a poset $(P, \leq_P)$

procedure VR(Integer k)

```
1: if k > n then
2:    print π
3: else
4:    VR(k + 1)
5:    i ← k
6:    while π_{i−1} ||_P π_i do
7:       swap the elements π_{i−1} and π_i
8:       VR(k + 1)
9:       i ← i − 1
10:   rotate left (π_i, π_{i+1}, ..., π_k)
```

More recently, Pruesse and Ruskey presented an algorithm [102] that always generates all linear extensions in $\mathcal{O}(e(P))$ time. On average, this algorithm only requires constant processing time for each linear extension, and are optimal up to a constant factor; such an algorithm is said to generate a linear extension in *constant amortized time*. Moreover, so-called loopless algorithms, which require even in the worst case only a constant processing time per linear extension, have been developed by Canfield and Williamson [26] and by Ono and Nakano [95].

Remark 3.2.1. A lot of research has been done in listing linear extensions in such a way that successive extensions only differ in a small, prescribed way. Such listings are called *Gray codes*. This problem of generating linear extensions by (adjacent) transpositions has been formalized by Ruskey in [105], and since then deepened [100, 106, 107, 112, 119].

Remark 3.2.2. In some cases one is interested in generating linear extensions with a minimal number of jumps [8, 90, 114]. The *jump number* of a linear extension of $(P, \leq_P)$ is defined as the number of times that two consecutive elements in that linear extension are incomparable in $(P, \leq_P)$.

3.2.1.2 Counting all rankings

Determining the number of linear extensions $e(P)$ of a given poset P is known to be a hard problem. In 1991, Brightwell and Winkler [16] have

shown it to be a #P-complete problem, thereby settling a long-standing open problem in order theory. The *#P-complete* complexity class denotes a class of counting problems similar to the NP-class for decision problems. It is generally believed that there exists no polynomial-time algorithm for solving #P-complete problems. Counting the number of linear extensions of a partially ordered set is therefore possibly not considerably easier than the enumeration of its linear extensions.

Recently, an algorithm for calculating $e(P)$ was suggested by Peczarski [97], which is, not surprisingly, not polynomial in the number of elements of the poset. The main idea of the algorithm of Peczarski consists of recursively applying the following Theorem 3.2.4.

Definition 3.2.3 (Peczarski [97])**.** Let $(P, \leq_P)$ be a poset, $D \subseteq P$, $a, b \in P$ and $d \in D$. The pair (A, B) is called an *admissible partition* of D with respect to the element d when the following conditions are satisfied:

- $A \cup B = D \setminus \{d\}$ and $A \cap B = \emptyset$
- $a <_P d \;\Rightarrow\; a \in A, \forall\, a \in D \setminus \{d\}$
- $d <_P b \;\Rightarrow\; b \in B, \forall\, b \in D \setminus \{d\}$
- $b \not<_P a$, $\forall\, a \in A,\ b \in B$

$\diamond$

Theorem 3.2.4 (Peczarski [97])**.** *Let $(P, \leq_P)$ be a poset.*

1. *If $A, B \subseteq P$, $A \cap B = \emptyset$ and $a \,||_P\, b$ for any $a \in A$ and $b \in B$, then*

$$e(A \cup B) = e(A) \cdot e(B) \cdot \binom{|A| + |B|}{|A|} = e(A) \cdot e(B) \cdot \binom{|A| + |B|}{|B|}.$$

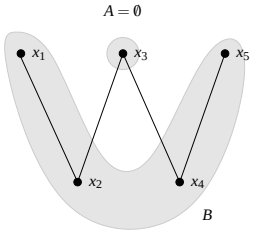
2. *If $D \subseteq P$ and $d \in D$, then*

$$e(D) = \sum_{A,B} e(A) \cdot e(B)$$

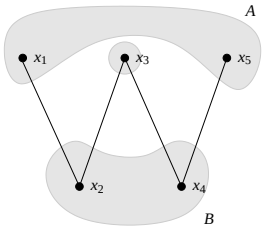
where the sum is taken over all admissible partitions of D with respect to d, and where the convention is made that $e(\emptyset) = 1$.

First, a poset $(P, \leq_P)$ is partitioned into connected subposets, *i.e.* posets that are a subset of $(P, \leq_P)$ and have a connected Hasse diagram. The number

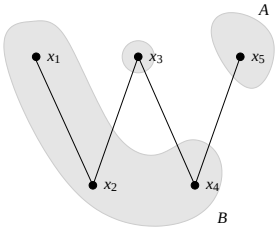
of linear extensions of the subposets can be computed independently and afterwards combined using the first part of Theorem 3.2.4. For each connected subposet, the second part of the theorem is used to obtain a set of simpler subproblems. Care is taken that an element d is chosen such that the number of admissible partitions is the smallest possible. It turns out that by subdividing the problem in this way the number of comparable elements in each subposet is maximal, substantially reducing the computation time.



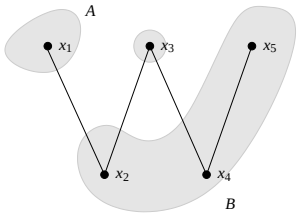
$A = \emptyset,\ e(A) = 1$
$B = \{x_1, x_2, x_4, x_5\},\ e(B) = 6$



$A = \{x_1, x_5\},\ e(A) = 2$
$B = \{x_2, x_4\},\ e(B) = 2$



$A = \{x_5\},\ e(A) = 1$
$B = \{x_1, x_2, x_4\},\ e(B) = 3$



$A = \{x_1\},\ e(A) = 1$
$B = \{x_2, x_4, x_5\},\ e(B) = 3$

Figure 3.1: An example poset $(X, \leq_X)$ with its admissible partitions when choosing $d = x_3$.

Example 3.1. In Figure 3.1 an example poset $(X, \leq_X)$ with $X = \{x_1, x_2, x_3, x_4, x_5\}$ is partitioned in its admissible partitions when choosing $d = x_3$. Applying the second part of Theorem 3.2.4 allows us to recursively compute the number of linear extensions of $(X, \leq_X)$, such that we obtain

$$e(X) = 1 \cdot 6 + 2 \cdot 2 + 1 \cdot 3 + 1 \cdot 3 = 16.$$

•

Remark 3.2.5. Algorithms to approximate $e(P)$ of a given poset P have been introduced in literature as well. Ewacha *et al.* [55] approximate $e(P)$ by

determining successive lower and upper bounds for $e(P)$ by counting so-called critical suborders. Moreover, Markov Chain Monte Carlo methods allow one to sample uniformly from the set of linear extensions of a poset. Such a method will be discussed in Paragraph 3.2.3.2. The ability to sample (almost) uniformly from this set implies that good approximations for $e(P)$ can be obtained.

Remark 3.2.6. Brightwell *et al.* [15] have found an asymptotic formula for the average number of linear extensions of an n-element poset, which is given in Theorem 3.2.7. Note the exponential nature in n of the asymptotic expressions.

Theorem 3.2.7 (Brightwell *et al.* [15]). *Asymptotically, the average number A_n of linear extensions of an n-element poset is given by*

$$A_n = \begin{cases} \dfrac{\chi^2 \cdot \phi_2}{2^{5/4} \cdot \phi_1} \cdot \left(\dfrac{n!}{2}\right)^2 \cdot n \cdot 2^{-n/2} \cdot \left(1 + \mathcal{O}\left(C^{-n}\right)\right) & \\ \simeq 5.041445\ldots \cdot \left(\dfrac{n!}{2}\right)^2 \cdot 2^{-n/2} & \text{if } n \text{ even} \\ \dfrac{\chi^2 \cdot \phi_2}{2^{5/4} \cdot \phi_1} \left(\dfrac{(n-1)!}{2}\right) \left(\dfrac{(n+1)!}{2}\right) \cdot n \cdot 2^{-n/2} \cdot \left(1 + \mathcal{O}\left(C^{-n}\right)\right) & \\ \simeq 5.041419\ldots \cdot \left(\dfrac{(n-1)!}{2}\right) \cdot \left(\dfrac{(n+1)!}{2}\right) \cdot n \cdot 2^{-n/2} & \text{if } n \text{ odd} \end{cases}$$

where

$$\begin{aligned} \phi_1 &= \sum_{j=-\infty}^{+\infty} 2^{-(j+1/2)^2} = \sqrt{\frac{n}{\ln 2}} \cdot \left[1 + 2 \cdot \sum_{n=1}^{+\infty} (-1)^n \cdot \exp\left(\frac{-\pi^2}{\ln 2} \cdot n^2\right)\right] \\ &= 2.1289312\ldots \\ \phi_2 &= \sum_{j=-\infty}^{+\infty} 2^{-j^2} = \sqrt{\frac{n}{\ln 2}} \cdot \left[1 + 2 \cdot \sum_{n=1}^{+\infty} \exp\left(\frac{-\pi^2}{\ln 2} \cdot n^2\right)\right] \\ &= 2.1289368\ldots \\ \chi &= \sum_{m=0}^{+\infty} p(m) \cdot 2^{-m} = \prod_{i=1}^{+\infty} \frac{1}{1 - 2^{-i}} = 3.4627466\ldots \end{aligned}$$

3.2.2 Using the lattice of ideals representation

In this section we will present our algorithm to generate uniformly at random one or more rankings of the elements of a poset [43]. The algorithm is based on the lattice of ideals representation of the poset and does not require enumerating all possible rankings. It consists of two independent parts. The first part builds up a data structure allowing for the fast generation of random rankings of the elements, *i.e.* linear extensions of a poset $(P, \leq_P)$, producing the number of linear extensions of $(P, \leq_P)$ as a side-result. The second part consists of the generation of a random linear extension itself.

3.2.2.1 Building the data structure

Since we want to be able to generate multiple random extensions without major additional cost, we would like to have a data structure at our disposal allowing for the quick selection of successive elements in the linear extension. The Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ is an adequate choice for this, since we know from Lemma 2.2.3 that a linear extension corresponds to a path from the source $v_\perp$ to the sink $v^\top$. In Subsection 2.3.2 an efficient algorithm developed by Habib *et al.* [71] to construct L_P by using an intermediate tree of ideals T_P is presented. We will extend their data structure representing L_P, and store for each vertex $v \in V_{L_P}$ the following data:

- $\mathrm{LEF}(v)$, the number of linear extensions of the filter $P \setminus I_f(v)$ of P where $I_f(v)$ yields the ideal corresponding to vertex v;
- $\mathrm{Visited}(v)$, a boolean flag indicating whether vertex v has already been visited by the algorithm.

Algorithm 3.2 invokes the procedure Assign with the source $v_\perp$ of L_P as argument, corresponding to the filter $P \setminus I_f(v_\perp) = P$. After recursion, the variable e in line 3 contains the number of linear extensions of $(P, \leq_P)$. Also, to each vertex $v \in V_{L_P}$ is associated the number of linear extensions of the complementary filter $P \setminus I_f(v)$. Note that dynamic programming is used in order to avoid repeated computation of $\mathrm{LEF}(v)$ for vertices $v \in V_{L_P}$.

Algorithm 3.2 Extending the Hasse diagram L_P of the lattice of ideals of a poset $(P, \leq_P)$ with counting information

```
1: initialize an array Visited setting its components to false
2: build the Hasse diagram L_P of the lattice of ideals of (P, ≤_P)
3: e ← Assign(v_⊥)

procedure Assign(Vertex v)
1: Visited(v) ←true
2: e ← 0
3: for each vertex v' ∈ ImSucc(v) do
4:    if v' = v^⊤ then
5:       e ← 1
6:    else
7:       if not Visited(v') then
8:          e ← e + Assign(v')
9:       else
10:         e ← e + LEF(v')
11: LEF(v) ← e
12: return e
```

Example 3.2. In Figure 3.2 the Hasse diagram L_Ω of the lattice of ideals of our example poset $(\Omega, \leq_\Omega)$ is shown with the counting information obtained by applying Algorithm 3.2. •

Since every edge in L_P is visited exactly once in the recursive algorithm, it is clear that the time complexity is linear in the size of the lattice of ideals, therefore not adding to the asymptotic time complexity of the construction algorithm of Habib *et al.* [70] in Theorems 2.3.2 and 2.3.3.

3.2.2.2 Generating a ranking uniformly at random

As stated in Lemma 2.2.3, a linear extension corresponds to a path in L_P from $v_\perp$, corresponding to the empty ideal, to $v^\top$, corresponding to the maximum ideal P. Therefore, the problem of generating a linear extension uniformly at random is reduced to the problem of sampling uniformly a maximum-length path in L_P. Let us assume that part of the random path in L_P has been generated up to a vertex $v_{k-1} \in V_{L_P}$, and that at this point in

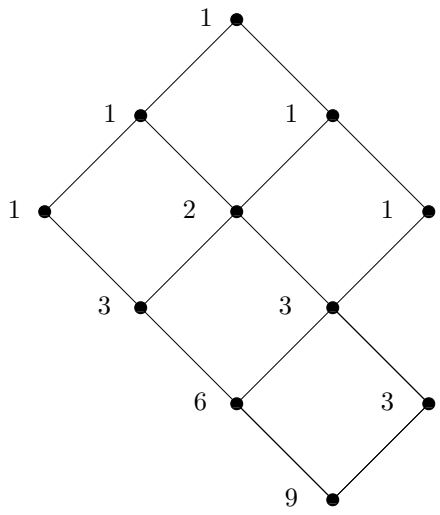


Figure 3.2: The Hasse diagram L_Ω of the lattice of ideals of $(\Omega, \leq_\Omega)$ with for each vertex $v \in V_{L_\Omega}$ the number of linear extensions of the filter $\Omega \setminus I_f(v)$ obtained from the application of Algorithm 3.2.

the path for vertex v_{k-1} a successor vertex v_k from $\{v_{i_1}, v_{i_2}, \ldots, v_{i_p}\} \subseteq V_{L_P}$ can be chosen. As already mentioned, the data structure constructed in the previous section contains the number of linear extensions in the complementary filter $P \setminus I_f(v_k)$. From this observation it immediately follows that the probability that should be assigned to a certain choice $v_k \in V_{L_P}$, equals $\mathrm{LEF}(I_f(v_k))/\mathrm{LEF}(I_f(v_{k-1}))$.

Since all ratios $\mathrm{LEF}(I_f(v_i))/\mathrm{LEF}(I_f(v_{i-1}))$, for $i = 2, \ldots, |P|$, have to be calculated for every element in the poset $(P, \leq_P)$, the time complexity for generating a linear extension of $(P, \leq_P)$ using the data structure from Algorithm 3.2 equals $\mathcal{O}(|P| \cdot w(P))$. The problem is now reduced to uniform sampling in a set of p alternatives with given probabilities. The pseudocode of this algorithm is shown in Algorithm 3.3.

Note that the function randLong in line 6 of Algorithm 3.3 returns a random number between zero and the largest number representable. The notation $\sigma + I_f(v') \setminus I_f(v)$ in line 10 means that the linear extension is extended with a maximal element $I_f(v') \setminus I_f(v)$, and the symbol $()$ in line 1 denotes an empty extension. Remark that the linear extension that is generated

uniformly at random is gradually built up in σ and then returned.

Example 3.3. In Figure 3.3 the Hasse diagram L_Ω of the lattice of ideals of $(\Omega, \leq_\Omega)$ is shown where part of the path has been generated with Algorithm 3.3. After the elements ω_1 and ω_2 are chosen, a probability of $2/3$ is attributed to the choice of ω_3 as third element, while a probability of $1/3$ is attributed to choosing ω_4. •

Special care should be taken in an implementation of the assignment algorithm. Since the number of linear extensions of a poset can be huge, a slightly modified version that only saves approximate values for $\mathrm{LEF}(v)$ where $v \in V_{L_P}$ could be suggested. Instead of saving integer values, a transformation into real values could be introduced when a certain threshold is exceeded. Only the exponent and some digits of the mantissa are then saved. Since the numbers of extensions added up in general have the same order of magnitude, the approximation error will be negligible. Remark that in this case only an approximate result for the number of linear extensions will be obtained and that a slight bias on the distribution will be induced. However, since in the context of the random generation of linear extensions we are interested in ratios, for the above mentioned reason, we can ignore this in practice.

Algorithm 3.3 Generating one linear extension of a poset $(P, \leq_P)$ on the basis of the data structure generated by Algorithm 3.2

```
 1: σ ← ()
 2: v ← v_⊥
 3: while ImSucc(v) ≠ ∅ do
 4:    t ← LEF(v)
 5:    c ← 0
 6:    r ← 1 + randLong() mod t
 7:    for each v' ∈ ImSucc(v) do
 8:       c ← c + LEF(v')
 9:       if r ≤ c then
10:          σ ← σ + I_f(v') \ I_f(v)
11:          v ← v'
12:          break for
13: return σ
```

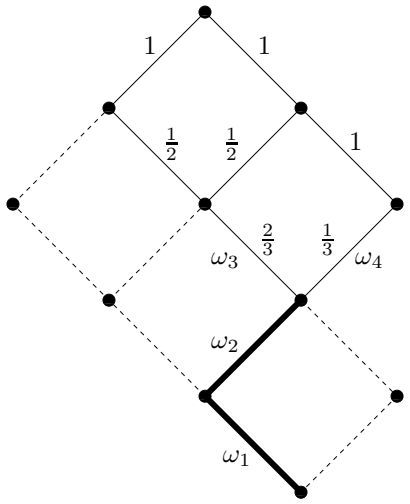


Figure 3.3: The Hasse diagram L_Ω where part of the path has already been generated.

3.2.2.3 A note on the time complexity

For an arbitrary poset, the number of ideals $i(P)$ is in general much lower than the number of linear extensions $e(P)$. Table 3.1 gives some evidence for this statement. We have generated 1 000 posets of different size n by choosing each time uniformly at random n points (x_1, x_2) out of a two-dimensional grid of size 20 by 20, equipped with the usual product ordering. In the table the average number of linear extensions and the average number of ideals is shown for each n. Note that when choosing random points from a higher dimensional space, the difference between the numbers of linear extensions and ideals is even expected to grow.

It should be noted that $i(P)$ is still exponential in the size of the poset. Indeed, consider the extreme case of an antichain poset where the number of ideals is 2^n for an antichain consisting of n elements. However, this is still clearly better than its number of linear extensions, which equals $n!$.

Fast algorithms enumerating all ideals of a poset are known. However, the problem of finding an algorithm generating all ideals in constant amortized time is still an open question. The fastest currently known algorithm enu-

merating all ideals of a poset is due to Squire [111] and generates all ideals in $\mathcal{O}(\log n)$ amortized time. For certain subclasses of posets, such as series-parallel posets or forest posets, better algorithms are available [81, 101]. Similar to counting all linear extensions, counting all ideals of a poset is shown to be #P-complete [53]. The problem of determining $i(P)$ even resides in the subclass #RHΠ_1 of #P, which can be considered as being the hardest subclass of #P to approximate.

n	$e(P)$	$i(P)$
5	12	11
10	$2.31\ 10^3$	51
15	$1.34\ 10^6$	158
20	$2.23\ 10^9$	403
25	$4.28\ 10^{12}$	933
30	$8.52\ 10^{15}$	2023
35	$3.39\ 10^{19}$	3808

Table 3.1: The average number of ideals compared to the average number of linear extensions for 1 000 generated posets with n elements.

3.2.3 Approximate sampling

3.2.3.1 The approach of Lerche and Sørensen

The approach of Lerche and Sørensen [83] consists in selecting a random pair of incomparable elements in the poset and imposing a random order on these elements. This procedure is repeated until no incomparable elements are left. Their algorithm generates a single linear extension in polynomial time in the number of elements of the poset. However, it does not guarantee that every extension is generated with equal probability and neither does it guarantee that the probabilities are close to being equal.

3.2.3.2 A Markov chain Monte Carlo method

Markov chain Monte Carlo methods (often abbreviated as MCMC methods) provide an algorithm for the following general computational task. Let $\mathcal{X}$ be a very large but finite set of combinatorial structures, and let π be a probability distribution on $\mathcal{X}$. Now sample an element of $\mathcal{X}$ at random, according to the distribution π. This is precisely the kind of task we want to solve: the set of structures $\mathcal{X}$ is the set of linear extensions $\mathcal{E}(P)$ of a poset $(P, \leq_P)$ and the probability distribution π is the uniform distribution.

Markov chain Monte Carlo methods have been used for many years and in several application areas like combinatorial optimization and computational physics. Frequently, these algorithms are heuristic in nature. However, recently analytical tools have been developed allowing some of these algorithms to have precise performance guarantees.

Different approaches [25, 52, 89] based on the Markov chain Monte Carlo method are capable of sampling almost uniformly from $\mathcal{E}(P)$ of a given poset $(P, \leq_P)$ with some given accuracy. In this section, we will describe an algorithm with a so-called rapidly mixing Markov chain developed by Bubley and Dyer [25].

Without loss of generality, we assume that $P = \{1, 2, \ldots, n\}$. Let us denote as $\sigma(i, j)$ the transposition operator on complete orders, such that if $Y = \sigma(i, j)X$, for

$$X = (a_1, a_2, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{j-1}, a_j, a_{j+1}, \ldots, a_n),$$

we have that

$$Y = (a_1, a_2, \ldots, a_{i-1}, a_j, a_{i+1}, \ldots, a_{j-1}, a_i, a_{j+1}, \ldots, a_n),$$

where $a_i \in P$ for all $i \in \{1, 2, \ldots, n\}$. Note that $a_k < a_l$ for all $a_k, a_l \in P$ if a_k is ranked before a_l in the list. For a given concave probability distribution f on $\{1, 2, \ldots, n-1\}$, define a Markov chain $\mathcal{M}_f$ on $\mathcal{E}(P)$ as follows. If the current state is $X_t \in \mathcal{E}(P)$ at a given time $t \geq 0$, then the next state X_{t+1} is determined by the following experiment:

1: choose $p \in \{1, 2, \ldots, n-1\}$ according to the distribution f
2: choose $c \in \{0, 1\}$ uniformly at random
3: **if** $c = 0$ **or** $\sigma(p, p+1)X_t \notin \mathcal{E}(P)$ **then**

4: $X_{t+1} \leftarrow X_t$
5: **else**
6: $X_{t+1} \leftarrow \sigma(p, p+1)X_t$

It is easily seen that $\mathcal{M}_f$ is *ergodic*, *i.e.* each state is reachable from an arbitrary state (but not necessarily in one move) with a non-zero probability. Moreover, $\mathcal{M}_f$ has a uniform *stationary distribution* since it is a symmetric chain: the probability of moving from state X_t to X_{t+1} is identical to the probability of moving from state X_{t+1} to X_t. These are sufficient conditions for the probability distribution on $\mathcal{E}(P)$ to converge towards the uniform distribution if $t \to \infty$.

A theoretical bound for the running time necessary to obtain some precision measure ϵ defined in Definition 3.2.9 is given in Theorem 3.2.11.

Definition 3.2.8. The *total variation distance* d_{TV} over a space $\mathcal{X}$ is defined as a function $d_{TV} : (P, Q) \to \frac{1}{2}\sum_{x\in\mathcal{X}} |P(x) - Q(x)|$ for two probability distributions P and Q. ◇

Definition 3.2.9. The *precision* ϵ of a Markov chain $\mathcal{M}$ is defined as an upper bound for the total variation distance d_{TV} between the observed probability distribution and the stationary distribution. ◇

Definition 3.2.10. The *mixing time* $\tau(\epsilon)$ of a Markov chain $\mathcal{M}$ is the simulation time, *i.e.* the number of chain transitions, required to obtain a precision of ϵ. A chain $\mathcal{M}$ is said to be *rapidly mixing* if $\tau(\epsilon) = \mathcal{O}(\text{poly}(\log(N/\epsilon))$, with N the number of states. ◇

Remark that in our case, the number of states N is exponential in the number of elements n in $(P, \leq_P)$. Therefore, a chain is rapidly mixing if we need to simulate the chain only for a number of steps that is polynomial in n in order to get a good sample from $\mathcal{E}(P)$.

Theorem 3.2.11 (Bubley and Dyer [25])**.** *When f is defined as $f : i \to i(n-i)/K$ where $K = (n^3 - n)/6$, $\mathcal{M}_f$ has a mixing time of $\mathcal{O}(n^3 \log n\epsilon^{-1})$, where ϵ is the precision of $\mathcal{M}_f$.*

As a last remark we should note that recently an algorithm based on the Markov chain Monte Carlo method has been developed by Huber [73] that generates perfectly uniformly distributed linear extensions. This algorithm uses non-Markovian coupling combined with a modified form of coupling

from the past, and has an expected running time of $\mathcal{O}(n^3 \log n)$. We should remark that, although the running time concentrates strongly around the expected running time, it still has a probabilistic nature. Moreover, the same procedure has to be repeated for each linear extension that needs to be generated.

3.3 Generating monotone data sets

Recall that a monotone data set consists of a poset of objects $(O, \leq_O)$, a set of labels L that is linearly ordered, and a monotone classification $d : O \to L$. A poset of objects $(O, \leq_O)$ can be obtained by sampling n elements from $\mathbb{R}^k$ equipped with the natural product ordering. The marginal distribution imposed on $\mathbb{R}^k$ can be chosen a priori. A set L of unique labels has to be chosen a priori as well.

Now we are left with the problem of generating a monotone classification $d : O \to L$. The specific case where as many labels as objects are present has been covered in Section 3.2. Indeed, in this case a random monotone classification can be obtained by sampling uniformly at random a linear extension of $(O, \leq_O)$. In the general case where $|O| \geq |L|$, when there are more objects than labels, the generation of a random assignment of labels to objects corresponds to the random generation of a weak order extension. Informally, a weak order extension $(P, \leq_W)$ of a poset $(P, \leq_P)$ can be regarded as a linear order on the equivalence classes (hereafter called classes for short) of a partition of $(P, \leq_P)$ not contradicting the underlying order. Objects residing in the same class will then be assigned the same label. To reduce the complexity of the problem, we can a priori fix the cardinality of each of the equivalence classes, *i.e.* we can specify the number of objects that will map to each label beforehand. In this section, we will adopt this type of restriction and therefore tackle the problem of the random generation of weak order extensions of a given poset with given *class cardinalities*. When generating a monotone data set, these class cardinalities can be fixed according to any distribution chosen.

Definition 3.3.1. A *weak order extension* $(P, \leq_W)$ of a poset $(P, \leq_P)$ is an extension of $(P, \leq_P)$ for which $\leq_W$ is a weak order. $\diamond$

Clearly, any weak order extension $(P, \leq_W)$ of a poset $(P, \leq_P)$ induces a par-

tition on P. Two elements x and y belong to the same class if and only if $x \leq_W y$ and $y \leq_W x$. Moreover, if they belong to different classes, then either $x \leq_W y$ or $y \leq_W x$. Since a linear order on the classes is defined, we can list the class cardinalities in that order as $(c_1, c_2, \ldots, c_k)$, where k denotes the number of classes. Remark that if $k = n$ the problem of generating a random weak order extension is equivalent to generating a random linear extension, which is described in the Section 3.2.

More formally, a weak order extension $(P, \leq_W)$ of $(P, \leq_P)$ with non-zero class cardinalities $(c_1, c_2, ..., c_k)$ can be characterized as a partition $(S_1, S_2, ..., S_k)$ of P for which it holds that

$$(\forall\, i \in \{1, 2, ..., k\})(|S_i| = c_i)\,,$$

$$(\forall\, i \in \{1, ..., k\})(\forall\, x, y \in S_i)(x =_W y)$$

and

$$(\forall\, i, j \in \{1, ..., k\})(\forall\, x \in S_i)(\forall\, y \in S_j)(i < j \Rightarrow x <_W y)\,,$$

where $x =_W y$ is equivalent with $x \leq_W y \wedge y \leq_W x$.

Remark 3.3.2. Remark that in the literature there is no real consensus on the definition of a weak order extension. Sometimes, it is preferred to regard a weak order extension $(P, \leq_W)$ still as a partition of P with a linear order on the classes, but with x and y belonging to the same class if neither $x \leq_W y$ nor $y \leq_W x$. The latter definition is not conform to the concept of an extension, but in some contexts it seems preferable to maintain the property of antisymmetry of the partial order $\leq_P$ at the expense of loosing its property of completeness (see [7]).

3.3.1 Known algorithms

An algorithm suggested by Lievens [85] generates a random weak order extension by traversing the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ from the source $v_\perp$ to the sink $v^\top$ in a pseudo-random way. When for an edge $e \in E_{L_P}$ on the path a successor edge e' has to be selected all possible successor edges $e' \in \mathrm{ImSucc}(e)$ are found and assigned equal probabilities. As a consequence, the whole Hasse diagram L_P need not be constructed. By using this simple greedy strategy, a linear extension of $(P, \leq_P)$ is obtained. This linear extension is subsequently converted into a weak order extension by partitioning according to the class cardinalities $(c_1, c_2, \ldots, c_k)$.

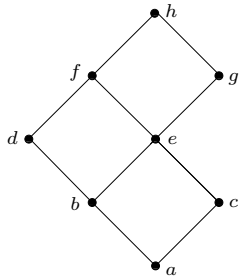


Figure 3.4: An example poset $(\Sigma, \leq_\Sigma)$.

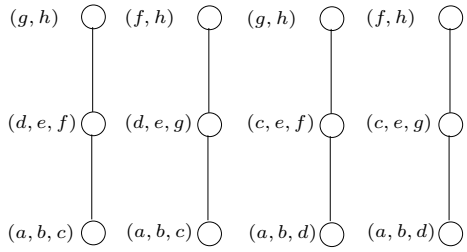


Figure 3.5: The weak order extensions of $(\Sigma, \leq_\Sigma)$ with class cardinalities $(3, 3, 2)$.

Potharst *et al.* [99] suggest the following procedure. First, the transitive closure $\bar{C}'_P$ of the Hasse diagram $C_P \cup \{x_D\}$ of $(P, \leq_P)$ is computed, where x_D is a dummy element such that $x_D > x$ for all $x \in P$. A list of not necessarily unique labels $(c_1, c_2, \ldots, c_n)$ to be assigned is sorted such that $c_1 \leq c_2 \leq \ldots \leq c_n$, where n is the number of elements in P. At this point the graph $\bar{C}'_P$ is interpreted as a flow network and each of the labels $c_1, c_2, \ldots, c_n$ is allowed to travel downward through the network from the vertex corresponding to the dummy element x_D in a random way. When a label comes to the end of its path, the label is attached to the last vertex on the path. This vertex is subsequently removed and the next label is allowed to travel through the new graph. This procedure is repeated until each vertex of the original graph has a label attached to it.

Example 3.4. In Table 3.2 the probability of each weak order extension outcome with both algorithms is shown, where the algorithm is applied to a new example poset $(\Sigma, \leq_\Sigma)$ shown in Figure 3.4. The poset $(\Sigma, \leq_\Sigma)$

has four weak order extensions with class cardinalities $(3,3,2)$ shown in Figure 3.5. From Table 3.2, it is immediately clear that both distributions are highly biased, even in this simple case where the poset is quite symmetrical. For example, with the algorithm described in [85], the second weak order extension listed will have a probability of being generated that is almost four times the probability of generating the third or fourth weak order extension. With the algorithm described in [99], the bias is even larger in this example.

Table 3.2: The generation probability of each weak order extension of the poset $(\Sigma, \leq_\Sigma)$ with class cardinalities $(3,3,2)$ with the algorithms in [85] and [99].

Weak order extension	Probability with [85]	Probability with [99]
$(a,b,c)\ (d,e,f)\ (g,h)$	28%	33.5%
$(a,b,c)\ (d,e,g)\ (f,h)$	47%	50.5%
$(a,b,d)\ (c,e,f)\ (g,h)$	12.5%	8%
$(a,b,d)\ (c,e,g)\ (f,h)$	12.5%	8%

•

3.3.2 Using the lattice of ideals representation

Consider for a given poset $(P, \leq_P)$ on the one hand all linear extensions of $(P, \leq_P)$, and on the other hand all weak order extensions of $(P, \leq_P)$ with given class cardinalities $(c_1, c_2, ..., c_k)$. Let us use the same class cardinalities to partition any linear extension such that the first c_1 elements in the linear order belong to the first class, the next c_2 elements to the second class, etc. In this way, we associate with each linear extension exactly one weak order extension. Note, however, that different linear extensions can map to the same weak order extension. Clearly, as it is our aim to generate weak order extensions uniformly at random [39], we would like to identify any weak order extension by means of a unique linear extension.

Example 3.5. For the example poset $(\Omega, \leq_\Omega)$ in Figure 2.2 the linear extensions are shown in Figure 3.6 and the weak order extensions corresponding with the class cardinalities $(2,2,1)$ are shown in Figure 3.7. For example, the four leftmost linear extensions in Figure 3.6 are all mapped to the leftmost weak order extension of Figure 3.7. Indeed, the four different linear extensions that are mapped to the same weak order extension differ only in the order of the elements residing in one and the same class. Remark that

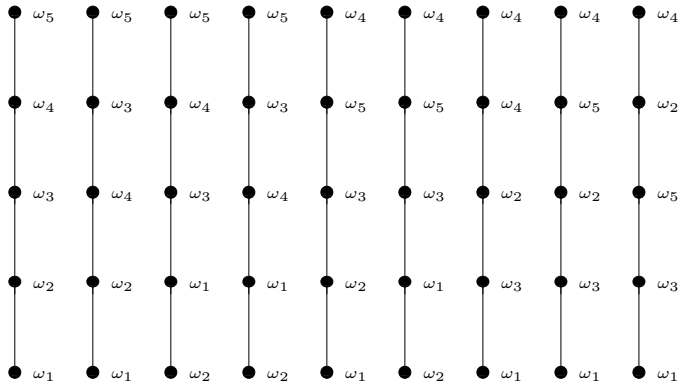


Figure 3.6: All linear extensions of $(\Omega, \leq_\Omega)$.

we have described the weak order extension according to the leftmost linear extension in Figure 3.6, *i.e.* the linear order where $\omega_i < \omega_j$ if and only if $i < j$ for all $i, j \in \{1, 2 \ldots, n\}$. •

3.3.2.1 Standardization of weak order extensions

We will introduce some lemmata that support a method to standardize weak order extensions, and will help to derive a recipe for assigning to each weak order extension a unique linear extension.

Lemma 3.3.3. *If $(P, \leq_L)$ is a linear extension of $(P, \leq_P)$ and $S \subseteq P$, then $(S, \leq_L)$ is a linear extension of $(S, \leq_P)$.*

The problem of generating random weak order extensions with given class cardinalities can be reduced to the random sampling from a subset of linear extensions that uniquely characterize the weak order extensions. To come up with a unique representation of a weak order extension, we first select any of the linear extensions of P which we denote from here onwards as $(P, \leq_L)$. Given a weak order extension $(P, \leq_W)$ of $(P, \leq_P)$, each class is labeled by its elements written in the order prescribed by $\leq_L$.

Example 3.6. This labeling for $(\Omega, \leq_\Omega)$ is illustrated in Figure 3.8 for the third weak order extension from the left in Figure 3.7 and with the lexicographic order fixed by the leftmost linear extension in Figure 3.6. •

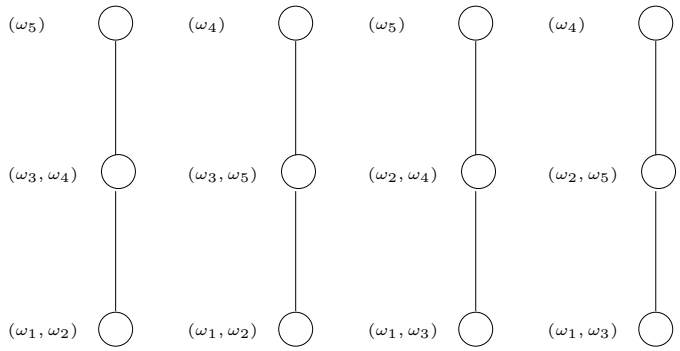


Figure 3.7: All weak order extensions of $(\Omega, \leq_\Omega)$ with class cardinalities $(2, 2, 1)$.

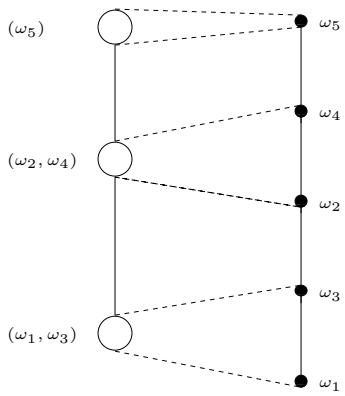


Figure 3.8: Weak order extension of $(\Omega, \leq_\Omega)$ with class cardinalities $(2, 2, 1)$ standardized by the leftmost linear extension in Figure 3.6.

More formally, given a fixed linear extension $(P, \leq_L)$ of $(P, \leq_P)$, we associate with each weak order extension $(P, \leq_W)$ a linear extension $(P, \leq_{L_W})$ such that $(\forall\, x, y \in P)(x =_W y \;\Rightarrow\; (x \leq_{L_W} y \Leftrightarrow x \leq_L y))$.

Lemma 3.3.4. *With the notations introduced above, for each choice of $\leq_L$ it holds that to each weak order extension $(P, \leq_W)$ of $(P, \leq_P)$ there corresponds a unique linear order $\leq_{L_W}$ on P.*

Proof : Given a fixed linear extension $(P, \leq_L)$ of P and any weak order extension $(P, \leq_W)$, let us consider the equivalence classes S_i induced by $\leq_W$. According to Lemma 3.3.3 the restriction of $\leq_L$ to S_i is a linear order on S_i, and this for all $i \in \{1, 2, ..., k\}$. These linear orders are concatenated into a linear order $\leq_{L_W}$ on P in agreement with the weak order $\leq_W$. The ordering of the classes S_i being unique in $\leq_W$ implies that the constructed linear order $\leq_{L_W}$ is unique too. ∎

Corollary 3.3.1. *On the basis of Lemmata 2.2.3 and 3.3.4, the problem of generating weak order extensions with given class cardinalities uniformly at random is reduced to the problem of sampling uniformly at random from the set of maximum-length paths of the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ for which it holds for every pair of subsequent edges that if their corresponding elements, say p and q, reside in the same class then necessarily $p \leq_L q$. In other words, all elements residing in a single class should be ordered according to a chosen linear extension $(P, \leq_L)$ of $(P, \leq_P)$.* ◇

3.3.2.2 Building the data structure

By taking the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ as the basic data structure, Corollary 3.3.1 now puts us in the position to design an appropriate algorithm for generating random weak order extensions with prescribed class cardinalities. Hence, we will assume that the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ is at our disposal (see Section 2.3) and use an approach similar to the one in Subsection 3.2.2.

The algorithm in Subsection 3.2.2 for generating a random linear extension essentially consists of two phases: a first phase in which the vertices of L_P are assigned an integer number, and a second phase in which a random path in the lattice is generated based on these vertex numbers. Recall that,

according to Lemma 2.2.3, a linear extension corresponds to a path in L_P from $v_\perp$ to $v^\top$. A vertex number counts the number of paths in the lattice from that particular vertex to $v^\top$, and therefore the generation of a random path uniformly from the pool of all possible paths is straightforward once the vertex numbers are available. For the present problem, we will follow the same line of thought and again construct a two-phase algorithm. In the first phase integer numbers will be assigned, this time, however, not to the vertices but to the edges of the lattice. The numbering of the edges proceeds by taking into account the constraints expressed in Corollary 3.3.1 which restricts the pool of all possible paths to the subset of those paths only that are in a one-to-one relationship to the weak order extensions of the poset. The second phase of our algorithm, on the other hand, closely resembles the one in our linear extension generating algorithm.

According to the condition mentioned in Corollary 3.3.1 it will be necessary to know whether two elements of $(P, \leq_P)$ corresponding to subsequent edges in a path of the lattice of ideals will belong to the same class of the weak order extension that is associated to that path. Hence, we introduce an array Class of length n where the value at index h indicates the label that should be assigned to an edge in L_P at height h, where the height is defined as the path length, *i.e.* the number of edges, from $v_\perp$ to the given edge. In fact, as it is convenient to use class labels $1, 2, \ldots, k$, the array Class is the concatenation of c_1 times 1, c_2 times 2, etc., where $(c_1, c_2, \ldots, c_k)$ has the same meaning as in the previous section. Furthermore, we will denote the element associated to an edge e as $\mathrm{Label}(e)$.

We will store for each edge $e \in E_{L_P}$ the following data:

- $\mathrm{Paths}(e)$, the number of valid paths from the edge e to $v^\top$;
- $\mathrm{Visited}(e)$, a boolean flag indicating whether edge e has already been visited by the algorithm;
- $\mathrm{Label}(e)$, the label of the edge e, *i.e.* the element of $(P, \leq_P)$ corresponding to the edge e.

The first phase of the weak order extension generating algorithm takes care of assigning the numbers to all the edges of the lattice of ideals and is listed in Algorithm 3.4. Remark that we assume that for each edge $e \in E_{L_P}$, a list $\mathrm{ImSucc}(e)$ of all successor edges in L_P is at our disposal.

Algorithm 3.4 Extending the Hasse diagram L_P of the lattice of ideals of a poset $(P, \leq_P)$ with counting information and returning the number of weak order extensions

```
1: initialize an array Visited setting its components to false
2: n ← 0
3: for each e′ ∈ ImSucc(e_⊥) do
4:    n ← n + Assign(e′, 0)
5: return n

function Assign(Edge e, Integer h)
1: Visited(e) ← true
2: m ← 0
3: for each e′ ∈ ImSucc(e) do
4:    if ImSucc(e′) = ∅ then
5:       Paths(e′) ← 1
6:       if Class(h) ≠ Class(h + 1) or Label(e) ≤_L Label(e′) then
7:          m ← m + 1
8:    else
9:       if not Visited(e′) then
10:         if Class(h) ≠ Class(h + 1) or Label(e) ≤_L Label(e′) then
11:            m ← m + Assign(e′, h + 1)
12:         else
13:            Assign(e′, h + 1)
14:      else
15:         if Class(h) ≠ Class(h + 1) or Label(e) ≤_L Label(e′) then
16:            m ← m + Paths(e′)
17: Paths(e) ← m
18: return m
```

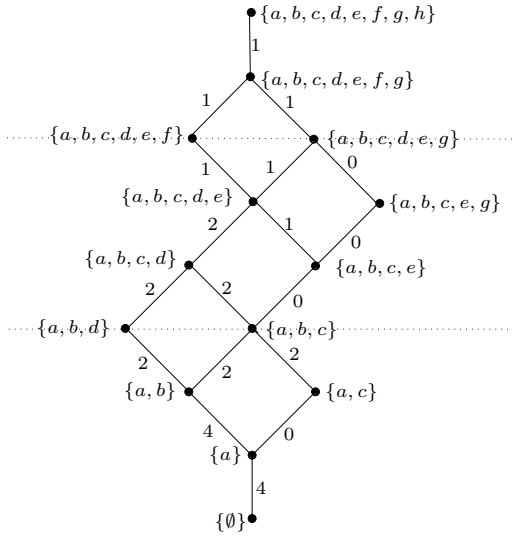


Figure 3.9: The Hasse diagram L_Σ of the lattice of ideals of $(\Sigma, \leq_\Sigma)$ extended with the counting information obtained by Algorithm 3.4.

Let us define a dummy edge $e_\perp$ between a dummy vertex v_0 and $v_\perp$ where $v_\perp = \text{Parent}(v_0)$. Furthermore define

$$\text{Paths}(e_\perp) = \sum_{e \in \text{ImSucc}(e_\perp)} \text{Paths}(e).$$

After running Algorithm 3.4, the total number of weak order extensions with the given class cardinality distribution is returned. Moreover, to every edge $e \in E_{L_p}$ the number of different paths in the lattice of ideals from e to $v^\top$ satisfying the condition of Corollary 3.3.1 is available as $\text{Paths}(e)$.

Example 3.7. Let us illustrate the numbering procedure on the example poset $(\Sigma, \leq_\Sigma)$ shown in Figure 3.4. All of its weak order extensions that are consistent with the class cardinalities $(3, 3, 2)$ and consistent with $\leq_L$, the lexicographic order, are depicted in Figure 3.5. Algorithm 3.4 establishes the numbering of L_Σ as shown on Figure 3.9. Dotted lines indicate the boundaries between classes. •

3.3.2.3 Generating a data set uniformly at random

The numbering procedure from the previous section will now substantially reduce the effort required to sample uniformly at random from the weak order extensions with given class cardinalities, actually reducing the problem to uniform sampling at random from a set of alternatives with given probabilities. Indeed, the random generation of a weak order is equivalent to the edgewise generation of a path from $v_\perp$ to $v^\top$ in the Hasse diagram L_T of the lattice of ideals of $(P, \leq_P)$ in the following way. Hypothesize the path is built up till height j and the last edge selected is e. For any possible successor edge e' it should either hold that $\text{Class}(j) \neq \text{Class}(j+1)$ or $\text{Label}(e) \leq_L \text{Label}(e')$. Edge e' is now selected with probability equal to the ratio between the numbers attached to e' and e. This is made explicit in Algorithm 3.5 listed hereafter. The algorithm generates a random weak order extension with prescribed class cardinalities.

Algorithm 3.5 Generating one weak order extension of a poset $(P, \leq_P)$ on the basis of the data structure generated by Algorithm 3.4

```
 1: σ ← ()
 2: e ← e_⊥
 3: h ← 1
 4: while ImSucc(e) ≠ ∅ do
 5:    t ← Paths(e)
 6:    c ← 0
 7:    r ← 1 + randLong() mod t
 8:    for each e' ∈ ImSucc(e) do
 9:       if Class(h) ≠ Class(h + 1) or Label(e) ≤_L Label(e') then
10:          c ← c + Paths(e')
11:          if r ≤ c then
12:             σ ← σ + Label(e')
13:             e ← e'
14:             h ← h + 1
15:             break for
16: return σ
```

Example 3.8. In Figure 3.10 the situation is shown in which only part of a random path has been generated so far. It is clear that, once arrived in the ideal $\{a, b, c\}$, only one next edge, *i.e.* the edge leading to $\{a, b, c, d\}$, can be chosen. Indeed, a probability 0 will be attributed to the edge leading

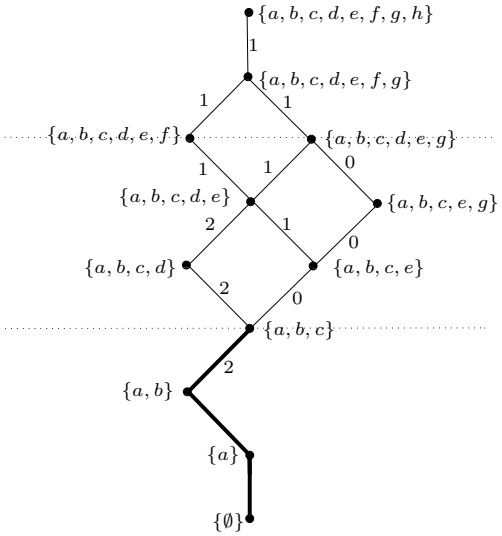


Figure 3.10: The extended Hasse diagram L_Σ of the lattice of ideals of $(\Sigma, \leq_\Sigma)$ where only part of the path has been generated.

to $\{a, b, c, e\}$ while a probability 1 will be attributed to the edge leading to $\{a, b, c, d\}$. •

By careful observation, it is immediately clear that the time complexity of the Algorithm 3.4 is $\mathcal{O}(w^2(P) \cdot i(P))$. The algorithm visits every edge in L_P once, and for each edge it visits each of its successors. Since the number of edges is $\mathcal{O}(w(P) \cdot i(P))$ and the number of successors is bounded by $\mathcal{O}(w(P))$, the stated complexity follows. It should again be noted that $i(P)$ could be exponential in the size of the poset, as explained in Paragraph 3.2.2.3.

Finally, we also want to point out the difference in time complexity between the present algorithm generating weak order extensions and the algorithm generating linear extensions established in Subsection 3.2.2. Using the present algorithm to generate a first random linear extension would require $\mathcal{O}(w^2(P) \cdot i(P))$ time, while using Algorithm 3.2 only requires $\mathcal{O}(w(P) \cdot i(P))$ time, *i.e.* a factor $w(P)$ less. The reason for this difference in time complexity lies in the difference of the numbering procedures. In general, when considering weak order extensions, not all successor edges of an edge in L_P are

acceptable as successors in the path. Moreover, two edges having the same set of successors in the lattice do not necessarily accept the same edges as successors. These facts force to number edges. Specializing to the particular case of linear extensions, all edges in L_P having the same successors accept all of these as successors in a linear order. This constant behavior allows to synthesize the information of successor edges into a single node number.

3.3.3 Approximate sampling

Since the time complexity of Algorithm 3.4 remains exponential in the number of elements of the object poset $(O, \leq_O)$, as argued in the previous section, this approach becomes infeasible for large posets, in particular for posets with a high number of incomparable elements. Therefore, an approximative approach is justified [41]. A lot of research has already been done on the almost uniform generation of linear extensions, and so-called rapidly mixing Markov Chain Monte Carlo methods have been developed [25, 52, 89]. Such an algorithm is discussed in Paragraph 3.2.3.2. However, to our knowledge, no algorithms have yet been suggested that allow for the almost uniform generation of weak order extensions with given class cardinalities $(c_1, c_2, \ldots, c_k)$.

Remark 3.3.5. Since, as described in the previous section, any weak order extension can be characterized by means of a unique linear extension once an initial linear order on the elements is fixed, one could suggest to use one of the algorithms to (almost) uniformly generate a linear extension. Each time a linear extension is generated that does not correspond to the canonical representation of a weak order extension, it is skipped and a next one is generated. Although with this approach we are guaranteed of generating weak order extensions (almost) uniformly, when the number of weak order extensions is very small compared to the number of linear extensions (which, in practice, often is the case) this approach is clearly infeasible.

3.3.3.1 The algorithm

Define $g(l) = \sum_{i=1}^{l} c_i$ for $l \leq k$, let $\mathcal{E}(P)$ as usually denote the set of linear extensions of $(P, \leq_P)$ and let n be the cardinality of P. Furthermore, for a linear extension π let π_i denote the element at position i. First, we

generate an arbitrary linear extension $\sigma \in \mathcal{E}(P)$. Initially, set $X_0 = \sigma$. Now we define a Markov Chain $\mathcal{M}_f$ on $\mathcal{E}(P)$ as follows. Starting from a state $X_t \in \mathcal{E}(P)$, the next state X_{t+1} is determined by the following experiment:

1: $\pi \leftarrow X_t$
2: partition the linear extension π in subsets $S_1, \ldots, S_k$ such that $\pi_1, \ldots, \pi_{g(1)} \in S_1$ and $\pi_{g(i-1)+1}, \ldots, \pi_{g(i)} \in S_i$ for all $i = 2, \ldots, k$
3: choose $p \in \{1, ..., k-1\}$ and $r \in \{0, 1\}$ uniformly at random
4: **if** $r = 0$ **then**
5: $\quad X_{t+1} \leftarrow X_t$
6: **else**
7: $\quad$ choose $\pi_x \in S_p$ and $\pi_y \in S_{p+1}$ uniformly at random
8: $\quad$ move π_x from S_p to S_{p+1}, and move π_y from S_{p+1} to S_p
9: $\quad$ sort the elements of both S_p and S_{p+1} in π according to σ
10: $\quad$ **if** $\pi \notin \mathcal{E}(P)$ **then**
11: $\quad\quad X_{t+1} \leftarrow X_t$
12: $\quad$ **else**
13: $\quad\quad X_{t+1} \leftarrow \pi$

In order to show that $\mathcal{M}_f$ has a limiting stationary distribution it suffices to prove that $\mathcal{M}_f$ is ergodic. We will show next that each state is reachable from an arbitrary state (not necessarily in one move) with a non-zero probability. Note that, as self-transitions are possible, the chain $\mathcal{M}_f$ is then guaranteed to be aperiodic.

Theorem 3.3.6. *The chain $\mathcal{M}_f$ is ergodic.*

Proof : It is generally known that in the special case when $c_i = 1$ for all $i \in \{1, ..., k\}$, *i.e.* when precisely the linear extensions of $(P, \leq_P)$ are generated, $\mathcal{M}_f$ is an ergodic Markov chain [25]. In a directed graph, where we represent all states, *i.e.* all linear extensions, as vertices and all transitions as directed edges, this is equivalent to requiring a directed path between any two vertices. By no longer requiring the class cardinalities to be 1, and by sorting the elements of both S_p and S_{p+1} in π according to the initially chosen linear extension σ in each step, it is possible that groups of linear extensions are aggregated into a single weak order extension. In our directed graph representation, this means that groups of vertices might be substituted by a new vertex. Since in each step, $\pi_x \in S_p$ and $\pi_y \in S_{p+1}$ are chosen uniformly at random, the directed edges between two new vertices

will be the union of the edges between vertices of the corresponding groups. Clearly, since the union of the edges is taken, a directed path between any two vertices remains present, proving the ergodicity of $\mathcal{M}_f$.

Finally, it is easy to see that $\mathcal{M}_f$ is a symmetric chain, since $\pi_x \in S_p$ and $\pi_y \in S_{p+1}$ are chosen uniformly in each step and the cardinalities of S_p and S_{p+1} remain unchanged. Therefore, the limiting stationary distribution of $\mathcal{M}_f$ is the uniform distribution. ∎

Remark 3.3.7. The random selection of p in line 3 of the sampling algorithm is done uniformly. Theoretically, other distributions that attach a non-zero probability to each choice of p are possible as they do not compromise convergence towards uniform sampling of the set of weak order extensions. However, it is obvious that the rate of convergence can be influenced by this choice. In the extreme case where the number of classes equals n, it has been shown [25] that a well-chosen concave probability distribution function can improve theoretical upper bounds on what is called the mixing time of the Markov chain, which is a measure for the rate of convergence. As a heuristic, one could use such a probability distribution function. However, most probably due to the limited size of the posets we can consider in order to be still able to compute the number of weak order extensions exactly, we could not observe any clear improvement in performance.

3.3.3.2 Experiments

In this section some experiments are presented in order to illustrate the performance of the sampling algorithm in practice, and more precisely, to get an idea of the number of chain transitions in $\mathcal{M}_f$ necessary to obtain an acceptable precision. First, we draw 15 attribute vectors $(n = 15)$ uniformly at random from $L_1 \times L_2 \times L_3 \times L_4$, where $L_i = \{1, \ldots, 10\}$ for $i \in \{1, \ldots, 4\}$. For the set of labels $\{1, 2, 3\}$, three different distributions are chosen: (a) $(5, 5, 5)$, (b) $(4, 7, 4)$ and (c) $(3, 5, 7)$. Next, for each of the label distributions the exact number of weak order extensions is computed (see Table 3.3) with the algorithm using the lattice of ideals representation described in Subsection 3.3.2. Note that the number of elements n has been chosen such that this remains feasible.

	(a)	(b)	(c)
label distribution	$(5,5,5)$	$(4,7,4)$	$(3,5,7)$
weak order extensions	2.497	1.392	994

Table 3.3: The exact number of weak order extensions for label distributions (a), (b) and (c).

Experiment 1 In the first experiment the sampling algorithm is used to draw s samples from the set of weak order extensions with t chain transitions for each sample, for different values of s and t. First, we verify that all weak order extensions are generated a sufficient number of times (> 5). For this reason the results with $s = 10^3 n$ are omitted. Second, we applied a χ^2-test for each parameter combination to quantify the uniformity of the sampling. For each of the three label distributions (a), (b) and (c) this experiment is repeated 10 times, and the average p-values are shown in Tables 3.4–3.6. As can be seen from these tables, in this example an acceptable result is obtained when at least n^2 chain transitions are used for each weak order extension. Note that no significant differences for the three label distributions can be observed in this experiment.

$s \backslash t$	n	n^2	n^3	n^4
$10^4 n$	0,00024	0,58112	0,57797	0,56399
$10^5 n$	0,00001	0,64788	0,34950	0,57956

Table 3.4: The p-value for each test on the uniform distribution with t chain transitions and s samples with label distribution (a).

$s \backslash t$	n	n^2	n^3	n^4
$10^4 n$	0,00006	0,51077	0,67181	0,68937
$10^5 n$	0,00010	0,64519	0,52575	0,50538

Table 3.5: The p-value for each test on the uniform distribution with t chain transitions and s samples with label distribution (b).

Experiment 2 Let us denote by $\mathrm{Prob}(x < y)$ the probability that the label assigned to x is lower than the label assigned to y in a random weak order

$s\backslash t$	n	n^2	n^3	n^4
$10^4 n$	0,00006	0,27871	0,74072	0,63859
$10^5 n$	0,00006	0,47579	0,58410	0,49931

Table 3.6: The p-value for each test on the uniform distribution with t chain transitions and s samples with label distribution (c).

extension. Analogously, we write $\text{Prob}(x = y)$ for the probability that the same label is assigned to both x and y. We define the $A^2 \to [0,1]$ relation Q by $Q(x,y) = \text{Prob}(x < y) + \frac{1}{2}\text{Prob}(x = y)$. It is a reciprocal relation as it holds that $Q(x,y) + Q(y,x) = 1$ [38, 48, 49]. Reciprocal relations will be covered in more depth in Chapter 5. In the case of linear extensions, the relation Q represents the mutual rank probabilities [43], which are discussed in the next chapter in Section 4.3. Using our sampling algorithm, we approximate $Q(x,y)$, denoted as $\hat{Q}(x,y)$, by sampling (almost) uniformly from the set of weak order extensions. We use

$$\delta(Q,\hat{Q}) = \frac{1}{n^2} \sum_{(x,y)\in A^2} |Q(x,y) - \hat{Q}(x,y)|$$

as a measure for the approximation error. In Table 3.7 the approximation error $\delta(Q,\hat{Q})$ is shown for label distribution (a). Clearly, the more transitions and the more weak order extensions generated, the better the approximation is.

$s\backslash t$	n	n^2	n^3	n^4
$10^3 n$	0,00378	0,00140	0,00138	0,00127
$10^4 n$	0,00096	0,00055	0,00035	0,00033
$10^5 n$	0,00020	0,00010	0,00017	0,00011

Table 3.7: The approximation error $\delta(Q,\hat{Q})$ for t chain transitions and s samples.

4 Ranking the elements of a poset

4.1 Introduction

As we discussed in the informal introduction in Chapter 1 of this work, in many applications one would like to compare different objects and induce a ranking of them on the basis of some common criteria. Objects could be, for example, geographic regions or chemicals, criteria could be the degree of cadmium pollution of a region or the melting point of a chemical. In each case, an object will have some score for each criterion. One could, for example, consider the quantification of the pollution of each geographical region on the basis of different chemicals. Some region x might be characterized by a higher pollution of lead than another region y. At the same time, however, it might be the case that the same region x has a lower pollution of cadmium than region y. In this situation, according to these two criteria, it is impossible to compare region x to region y as long as no common scale among the criteria is fixed. Precisely the problem of finding such a common scale among the criteria of concern is surrounded with controversy. Instead of insisting on the construction of such a common scale, and thus the comparability of each object, one could instead opt to allow for incomparability between two objects. This observation is a motivation for using the theory of partially ordered sets for ranking objects in a multicriteria setting.

Partially ordered sets and their visualization by Hasse diagrams can be of

great help for a "data-driven" evaluation. One can find answers to questions such as: (i) what is the position of any object of interest; (ii) what is the relation of that object to the other ones (in terms of comparabilities or incomparabilities); (iii) why does one find an incomparability (see e.g. the concept of antagonistic indicators [109, 110]). However, if a decision is to be made, for example which region needs most urgently an improvement with respect to its pollution status, then the incomparabilities are often considered as an obstacle in finding such a priority list. Conventional approaches of Decision Support Tools like MAUT [108] or PROMETHEE [11], which can be considered as attempts to find a common scale (as mentioned above), are based on additional knowledge in modeling the preferences, for example by weights.

An alternative approach consists of linearizing the partially ordered set, which requires the selection of a linear extension. Indeed, typically the decision maker who wants to rank the elements of a poset needs to select a single linear extension out of a manifold of linear extensions. Algorithms for generating a linear extension uniformly at random have been discussed extensively in the previous chapter. The choice of the decision maker should, however, not be made arbitrarily. If, for example, a particular element has a low rank in most of the linear extensions, linear extensions in which that element occasionally appears at a higher rank should be discarded from the option list. The question thus arises which linear extension to select, if this makes sense at all. A popular approach [22] is to consider the linear extension with possible ties, *i.e.* the weak order extension, obtained by ranking the objects on the basis of their average ranks. These average ranks are based on the set of all linear extensions. However, as already pointed out in the previous chapter, straightforward counting or enumeration of all linear extensions becomes computationally intractable for partially ordered sets containing 15 objects or more. In Section 4.2 we present algorithms for computing these (average) rank probabilities, exactly as well as approximately, that aim to overcome this problem.

Instead of insisting on obtaining a ranking of all objects, in some cases knowing a probability that an object x is ranked higher than an object y can be sufficient or even preferable. The mutual rank probability of two elements x and y is used as an objective quantification for such a probability, since it expresses the probability that x is ranked higher than y in a linear extension sampled uniformly at random. A probability higher than $1/2$ suggests that it would be preferable to rank object x before object y. The higher this probability, the stronger the degree of preference of x over y. Algorithms

for deriving these mutual rank probabilities are described in Section 4.3.

We finally conclude this chapter with a real-world application of herb layer pollution in Baden-Württemberg in Germany to illustrate the application of the algorithms introduced in this chapter in order to rank partially ordered sets of objects.

4.2 Rank probabilities

Definition 4.2.1. The *rank probability* $\text{Prob}(\text{rank}(x) = i)$ of an element x of a poset $(P, \leq_P)$, with $1 \leq i \leq n$ where n is the cardinality of P, is defined as the probability that a linear extension $(P, \leq_L)$ of $(P, \leq_P)$ sampled uniformly at random from $\mathcal{E}(P)$ has $x >_L y$ for exactly $i-1$ elements $y \in P \setminus \{x\}$ and $x <_L y$ for all other elements $y \in P \setminus \{x\}$. ◇

Stated differently, the rank probability $\text{Prob}(\text{rank}(x) = i)$ is the fraction of linear extensions in which element $x \in P$ has rank i. Note that we say that an element has *rank* i in a linear extension if $i-1$ elements are smaller in that linear extension.

Definition 4.2.2. The *average rank* $\rho(x)$ of an element $x \in P$ is defined as the expected value of the rank of x, *i.e.*

$$\rho(x) = \sum_{i=1}^{n} i \cdot \text{Prob}(\text{rank}(x) = i).$$

◇

Since $\mathcal{E}(P)$ is the set of all rankings compatible with the underlying poset $(P, \leq_P)$, the average rank of an element gives a clear indication of how low or how high the element on average is ranked in all linear extensions. All rankings considered equally probable, for a decision maker a low average rank points to an increased probability that the element should receive a low rank, while a high average rank gives evidence the element should receive a high rank. Therefore, in some sense the average rank reflects a consensus.

Example 4.1. In Table 4.1 the rank probabilities $\text{Prob}(\text{rank}(x) = i)$ and average ranks $\rho(x)$ of the elements $x \in \Omega$ of the example poset $(\Omega, \leq_\Omega)$ from Figure 2.2 are shown. One can e.g. deduce that in 5 out of 9 cases,

element ω_5 receives rank 5, while it is never ranked on position 1 or 2. Remark furthermore that the average ranks of element ω_4 and ω_5 are close to each other, indicating that they behave quite similarly according to this measure, in contrast to e.g. the much wider gap between the elements ω_1 and ω_5. •

$x\backslash i$	1	2	3	4	5	ρ
ω_1	$2/3$	$1/3$	0	0	0	$4/3 \approx 1,333$
ω_2	$1/3$	$1/3$	$2/9$	$1/9$	0	$19/9 \approx 2,111$
ω_3	0	$1/3$	$4/9$	$2/9$	0	$26/9 \approx 2,889$
ω_4	0	0	$2/9$	$1/3$	$4/9$	$38/9 \approx 4,222$
ω_5	0	0	$1/9$	$1/3$	$5/9$	$40/9 \approx 4,444$

Table 4.1: The rank probabilities $\text{Prob}(\text{rank}(x) = i)$ and average ranks $\rho(x)$ of the elements x of $(\Omega, \leq_\Omega)$.

Note that it is possible for two or more elements of a poset to have an identical average rank. Moreover, the order in which two comparable elements occur in a poset is preserved in the average ranks of the elements. Therefore, if the elements are ranked according to their average rank, one obtains a linear order on equivalence classes induced by equal average ranks, and thus a weak order extension of the poset.

Example 4.2. In Example 4.1 the average ranks $\rho(x)$ of all $x \in \Omega$ are unique and induce a linear extension of $(\Omega, \leq_\Omega)$, *i.e.* $(\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$. •

4.2.1 Exact computation

First we describe how the exact distribution of the rank probabilities of a poset $(P, \leq_P)$ can be obtained [43]. A straightforward approach simply generates each linear extension of $(P, \leq_P)$ and counts for each element $x \in P$ the number of linear extensions in which x has rank $i \in \{1, ..., n\}$. As a next step these numbers are divided by the total number of linear extensions in order to obtain probabilities. The average rank $\rho(x)$ of each element $x \in P$ is then easily computed. For an overview of algorithms enumerating all linear extensions, we refer to Paragraph 3.2.1.1. An advantage of this approach clearly is its simplicity and minimal use of memory; its total memory requirement is only $\mathcal{O}(n^2)$. A huge drawback, however, is the fact that

all linear extensions need to be generated. As already described in Paragraph 3.2.2.3, for larger posets this approach becomes quickly infeasible due to the exponential behavior of the number of linear extensions.

We observe that counting the number of linear extensions of a poset $(P, \leq_P)$ with a specific element $x \in P$ on position i amounts to the problem of counting the number of paths in the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ containing the edges labelled x at height i, a problem that is easily solved by careful counting. The rank probability $\text{Prob}(\text{rank}(x) = i)$ then equals the fraction of the total number of paths, *i.e.* the number of linear extensions, containing an edge labelled x at height i. Recall that the height of an edge is defined as the path length from $v_\perp$ to this edge.

In order to compute for each edge the number of paths containing that edge, we suggest the following strategy. In Algorithm 3.2, for each vertex $v \in V_{L_P}$ the number of paths from $v^\top$ to v is computed and stored as $\text{LEF}(v)$. We can now apply a similar technique and compute the number of paths from v to $v_\perp$ and store this number as $\text{LEI}(v)$. The number of paths containing an edge $e \in E_{L_P}$ now equals the product of $\text{LEI}(v_t)$ and $\text{LEF}(v_h)$, where v_h is the head vertex of e and v_t its tail vertex.

Essentially, this means we will extend the data structure L_P and store for each vertex $v \in V_{L_P}$:

- $\text{LEF}(v)$, the number of linear extensions of the filter $P \setminus I_f(v)$ of P where $I_f(v)$ yields the ideal corresponding to vertex v;
- $\text{LEI}(v)$, the number of linear extensions of the ideal $I_f(v)$ of P;
- $\text{Visited}(v)$, a boolean flag indicating whether vertex v has already been visited.

The counting procedure is carried out in Algorithm 4.1. Remark that the algorithm consists of three procedures:

- AssignTopDown: the procedure used in Algorithm 3.2 to count the number of paths from $v^\top$ to each vertex $v \in V_{L_P}$.
- AssignBottomUp: L_P is traversed in a breadth-first way such that the number of paths from $v_\perp$ to each vertex $v \in V_{L_P}$ is computed.
- ComputeRankProb: the actual rank probabilities are computed on the

basis of the counting information.

Remark that the variable L in line 3 of the procedure AssignBottomUp represents an ordered list of vertices. The notation $L + v'$ adds a vertex v' to the end of the list L, and $L[0]$ in line 5 refers to the first vertex of L. In the procedure ComputeRankProb the two-dimensional array $rp[x,h]$ on line 9 equals $\text{Prob}(\text{rank}(x) = h)$ for $x \in P$ and $h \in \{1,2,\ldots,n\}$.

Example 4.3. The product of $LEI(v_t)$ and $LEF(v_h)$ for each edge $(v_t, v_h) \in E_{L_P}$ is shown in Figure 4.1. •

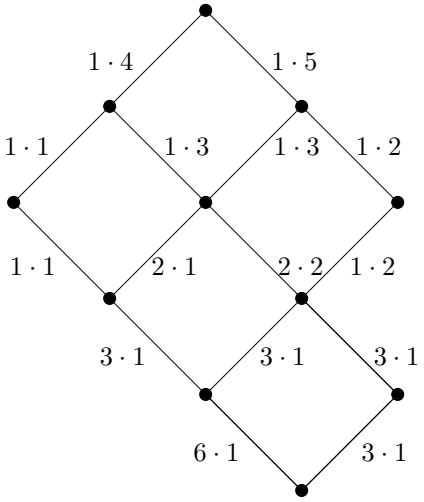


Figure 4.1: The Hasse diagram L_Ω of the lattice of ideals of $(\Omega, \leq_\Omega)$ with both path numbers assigned.

The above discussion implies that all rank probability distributions can be obtained by this two-pass technique in time linear in the number of edges in L_P, leading to a time complexity of $\mathcal{O}(i(P) \cdot w(P))$.

4.2.2 Approximate computation

Although the algorithm presented in the previous section will in practice perform much better than the approach consisting of enumerating all linear extensions, it still has a complexity that is exponential in the number of el-

Algorithm 4.1 Computing the rank probability distribution for each element x in $(P, \leq_P)$

initialize an array rp setting its components to 0
build the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$
initialize an array Visited setting its components to **false**
$e \leftarrow$AssignTopDown$(v_\perp)$
initialize an array Visited setting its components to **false**
AssignBottomUp()
initialize an array Visited setting its components to **false**
ComputeRankProb$(v_\perp, 0)$
return rp

procedure AssignTopDown(Vertex v)

Visited$(v) \leftarrow$**true**
$e \leftarrow 0$
for each vertex $v' \in$ ImSucc(v) **do**
  if $v' = v^\top$ **then**
    $e \leftarrow e + 1$
  else
    if not Visited(v') **then**
      $e \leftarrow e +$ AssignTopDown(v')
    else
      $e \leftarrow e +$ LEF(v')
LEF$(v) \leftarrow e$
return e

procedure AssignBottomUp()

LEI$(v_\perp) \leftarrow 1$
for each $v' \in$ ImSucc$(v_\perp)$ **do**
  $L \leftarrow L + v'$
while L **not empty do**
  $v \leftarrow L[0]$
  remove $L[0]$
  Visited$(v) \leftarrow$ **true**
  for each $v' \in$ ImSucc(v) **do**
    LEI$(v') \leftarrow$ LEI(v') + LEI(v)
    if not Visited(v') **then**
      $L \leftarrow L + v'$

Algorithm 4.1 (continued) Computing the rank probability distribution for each element x in $(P, \leq_P)$

procedure ComputeRankProb(Vertex v, Integer h)

```
1: Visited(v) ← true
2: for each v' ∈ ImSucc(v) do
3:    x ← I_f(v') \ I_f(v)
4:    rp[x,h] ← rp[x,h] + (LEI(v) × LEF(v')) /e
5:    if v' ≠ v^⊤ and not Visited(v') then
6:       ComputeRankProb(v', h + 1)
```

ements. Therefore, since no better algorithms are known, one has to resort to approximate algorithms for larger posets. In some cases a good approximation of the rank probabilities and average ranks can be acceptable.

Brüggemann *et al.* [23] suggest a simple formula to calculate an approximation of the average rank of an element $x \in P$ by using a so-called local partial order model. Their formula $\hat{\rho}(x) = (S(x)+1) \cdot (N+1)/(N+1-U(x))$ uses the number $S(x)$ of elements larger than x in P, the total number N of objects of P and the number $U(x)$ of objects incomparable with x in P. The main advantage of this approach is clearly its simplicity and the fact that it has a time complexity that is linear in the number of elements of P. They verify the quality of this approach by comparing the approximations with the exact average ranks for small example posets. Formulas for estimating the rank probabilities are also suggested. However, as it is easy to construct posets on which the approximations perform poorly, in some contexts they are of limited use.

Lerche *et al.* [83] and [84] use the approach described in Paragraph 3.2.3.1. It is used to obtain pseudo-random samples from the set of linear extensions $\mathcal{E}(P)$ to estimate the rank distributions and average ranks of $(P, \leq_P)$.

Furthermore, one could use the Markov chain Monte Carlo algorithm presented in Paragraph 3.2.3.2 to draw samples from $\mathcal{E}(P)$ (almost) uniformly at random. These samples can then be used to estimate the rank probabilities.

4.3 Mutual rank probabilities

Definition 4.3.1. The *mutual rank probability* $\text{Prob}(x > y)$ of two elements $x, y \in P$ for a poset $(P, \leq_P)$ is defined as the probability that for a linear extension $(P, \leq_L)$ of a poset $(P, \leq_P)$ sampled uniformly at random from $\mathcal{E}(P)$ it holds that $x >_L y$. ◇

Stated differently, the mutual rank probability $\text{Prob}(x > y)$ is the fraction of linear extensions in which element x is ranked higher than element y.

The number $\text{Prob}(x > y)$ can be seen as a degree of preference of ranking object x higher than object y. One can say that, the closer $\text{Prob}(x > y)$ to $1/2$, the more indifference there is between ranking x above y or y above x, while the closer $\text{Prob}(x > y)$ to 1, the higher the degree of preference of ranking x above y.

Example 4.4. In Table 4.2 the mutual rank probabilities $\text{Prob}(x > y)$ for each pair of elements x, y of $(\Omega, \leq_\Omega)$ are shown. One can e.g. see that there is a high degree of preference of ranking ω_5 above ω_2, while ω_4 and ω_5 are rather indifferent. •

$x \backslash y$	ω_1	ω_2	ω_3	ω_4	ω_5
ω_1	0	$1/3$	0	0	0
ω_2	$2/3$	0	$1/3$	0	$1/9$
ω_3	1	$2/3$	0	$2/9$	0
ω_4	1	1	$7/9$	0	$4/9$
ω_5	1	$8/9$	1	$5/9$	0

Table 4.2: The mutual rank probabilities $\text{Prob}(x > y)$ for each pair of elements x, y of $(\Omega, \leq_\Omega)$.

4.3.1 Exact computation

In order to obtain the exact mutual rank probabilities $\text{Prob}(x > y)$ for all pairs of elements x, y of a poset $(P, \leq_P)$ one can again use one of the enumeration algorithms for linear extensions of $(P, \leq_P)$ and count the number of linear extensions in which x is ranked higher than y. Also here, while it

has the advantage of needing only a limited amount of memory, it quickly becomes infeasible for larger posets.

Alternatively, the lattice of ideals representation of $(P, \leq_P)$ can be used to compute the mutual rank probabilities. We construct the lattice of ideals as described in Section 2.3 and we use Algorithm 4.1 from the previous section to obtain the number of paths from the source to the sink containing each edge of L_P. If we add all numbers associated to edges labelled y below an edge labelled x, we obtain the number of linear extensions for which it holds that x is ranked higher than y. Dividing this number by the total number of linear extensions gives the mutual rank probability $\text{Prob}(x > y)$. Of course, we insist on obtaining the mutual rank probabilities $\text{Prob}(x > y)$ for all $x, y \in P$ in a single pass, which will constitute a major gain in time compared to the standard approach of enumerating all linear extensions and calculating for every couple $x, y \in P$ the number of linear extensions obeying the condition $x > y$.

We will propose an algorithm that gradually builds up a two-dimensional table *mrp* initialized with zeroes, which will after execution of the algorithm contain the mutual rank probabilities $\text{Prob}(x > y)$ for all $x, y \in P$. The main idea is to traverse the lattice of ideals recursively in a depth-first manner, passing in each recursive call an array of boolean flags VisitedElement indicating which elements from P have already been visited in the traversal.

For each vertex $v \in V_{L_P}$ the following information will be stored:

- $\text{LEF}(v)$, the number of linear extensions of the filter $P \setminus I_f(v)$ of P where $I_f(v)$ yields the ideal corresponding to vertex v;
- $\text{LEI}(v)$, the number of linear extensions of the ideal $I_f(v)$ of P;
- $\text{Visited}(v)$, a boolean flag indicating whether vertex v has already been visited by the algorithm.

and for each element $x \in P$, we store:

- $\text{VisitedElement}(x)$, a boolean flag indicating whether element x has already been visited in the traversal.

The traversal of the Hasse diagram L_P is shown in Algorithm 4.2. Note that the procedures AssignTopDown and AssignBottomUp in lines 4 and 6 can

be found in Algorithm 4.1.

Algorithm 4.2 Computing the mutual rank probability for each couple $(x, y) \in P^2$ for $(P, \leq_P)$

```
 1: initialize an array mrp setting its components to 0
 2: build the Hasse diagram L_P of the lattice of ideals (I(P), ⊆) of (P, ≤_P)
 3: initialize an array Visited setting its components to false
 4: e ←AssignTopDown(v_⊥)
 5: initialize an array Visited setting its components to false
 6: AssignBottomUp()
 7: initialize an array Visited setting its components to false
 8: initialize an array VisitedElement setting its components to false
 9: ComputeMutualRankProb(v_⊥, 1)
10: return mrp

procedure ComputeMutualRankProb(Vertex v, Integer h)
 1: Visited(v) ← true
 2: for each v' ∈ ImSucc(v) do
 3:    for each y ∈ P do
 4:       if VisitedElement(y) then
 5:          x ← I_f(v') \ I_f(v)
 6:          mrp[x, y] ← mrp[x, y] + (LEI(v) × LEF(v'))/e
 7:    if v' ≠ v^⊤ and not Visited(v') then
 8:       VisitedElement(I_f(v') \ I_f(v)) ← true
 9:       ComputeMutualRankProb(v', h + 1)
10:       VisitedElement(I_f(v') \ I_f(v)) ← false
```

After the depth-first traversal the table $mrp[x, y]$ will precisely contain the mutual rank probabilities $\text{Prob}(x > y)$ for all $x, y \in P$ in line 10. It is easy to see that the required time complexity is bounded by $\mathcal{O}(i(P) \cdot n \cdot w(P))$ since for every edge of the lattice of ideals a loop over all the elements in P will be executed in line 3 of the procedure ComputeMutualRankProb.

4.3.2 Approximate computation

One could of course use the strategy from Subsection 4.2.2 and sample (almost) uniformly from the set of linear extensions to derive approximations

for the mutual rank probabilities. Alternatively, Brüggemann *et al.* [20] have developed a formula to approximate the mutual rank probabilities. They define $Q(x,y) = (N_u(x,y)+1)/(N_d(x,y)+1)$, where $N_u(x,y)$ is the number of objects above x which are not at the same time above y, and $N_d(x,y)$ is the number of objects under x which are not at the same time under y. The probability that x is ranked higher than y is then approximated as $Q(y,x)/(1+Q(x,y))$. A similar attempt to approximate mutual rank probabilities is described by Lerche *et al.* [84].

4.3.3 Relation between the average ranks and mutual rank probabilities

In Theorem 4.3.2 an interesting relation between the average ranks and mutual rank probabilities is pointed out. In order to obtain the average rank $\rho(x)$ of an element x of $(P,\leq_P)$ it is sufficient to compute the mutual rank probabilities $\text{Prob}(x > y)$ for all $y \in P$. As a result, it is not necessary to compute all rank probabilities $\text{Prob}(\text{rank}(x) = i)$ for $i = 1,\ldots,n$ to obtain the average rank $\rho(x)$ of x.

Theorem 4.3.2. *For a poset $(P,\leq_P)$ where $P = \{p_1,p_2,\ldots,p_n\}$ and $p_l \in P$, the following relation between the average ranks and the mutual rank probabilities holds:*

$$\rho(p_l) = \sum_{i=1}^{n} i \cdot \text{Prob}(\text{rank}(p_l) = i) = 1 + \sum_{j=1,\, p_j \neq p_l}^{n} \text{Prob}(p_l > p_j).$$

Proof : We will prove a slightly more general equality. Let A be any list of m permutations of n symbols $p_1,p_2,\ldots,p_n$. Denote by $n_i^{p_j}$ the number of times symbol p_j occurs at position i in A, and by $n_{p_j>p_i}$ the number of times symbol p_j occurs after symbol p_i in A.

We want to prove that for any symbol $p_l \in \{p_1,p_2,\ldots,p_n\}$ it holds that

$$\sum_{i=1}^{n} i \cdot n_i^{p_l} = m + \sum_{j=1,\, p_j \neq p_l}^{n} n_{p_l>p_j}. \tag{4.1}$$

Dividing both sides by m, the left-hand side denotes the average position of symbol p_l in A, whereas the right-hand side represents the sum of the ele-

ments in column l of the matrix B with elements $b_{ij} = \text{Prob}(p_j > p_i|A)$ for all $i, j \in \{1, 2, \ldots, k\}$ where $i \neq j$, and with $b_{ii} = 1$ for all $i \in \{1, 2, \ldots, n\}$.

The proof goes by induction. First let $n = 2$ and denote the 2 symbols as p_1 and p_2. Suppose A contains m_1 permutations (p_1, p_2) and $m_2 = m - m_1$ permutations (p_2, p_1). We have $n_1^{p_1} = n_2^{p_2} = m_1$, $n_2^{p_1} = n_1^{p_2} = m_2$, $n_{p_2 > p_1} = m_1$, $n_{p_1 > p_2} = m_2$ and $m_1 + m_2 = m$. Whence it holds that

$$\sum_{i=1}^{2} i \cdot n_i^{p_1} = n_1^{p_1} + 2 \cdot n_2^{p_1} = m_1 + 2 \cdot m_2 = m + m_2$$
$$\sum_{i=1}^{2} i \cdot n_i^{p_2} = n_1^{p_2} + 2 \cdot n_2^{p_2} = m_2 + 2 \cdot m_1 = m + m_1$$

yielding

$$\sum_{i=1}^{2} i \cdot n_i^{p_1} = m + n_{p_1 > p_2}$$

and

$$\sum_{i=1}^{2} i \cdot n_i^{p_2} = m + n_{p_2 > p_1}.$$

Therefore expression (4.1) is satisfied for $n = 2$.

Now suppose (4.1) is satisfied for some $n \geq 2$. We will now prove that this is also satisfied for $n + 1$. Let A be the given set of m permutations of $n + 1$ symbols. Denote any of these symbols as x and the remaining symbols as $p_1, p_2, \ldots, p_n$. With each permutation in A, define a new permutation of n symbols $p_1, p_2, \ldots, p_n$ by leaving out symbol x. Denote by A' the list of m permutations obtained in this way. Let us take any symbol $p_l \in \{p_1, p_2, \ldots, p_n\}$. Denote the number of times p_l occurs at position i in A' as ${n'}_i^{p_l}$ and the number of times p_l occurs after p_j in A' as ${n'}_{p_l > p_j}$. In A', equality (4.1) holds:

$$\sum_{i=1}^{n} i \cdot {n'}_i^{p_l} = m + \sum_{j=1,\, p_j \neq p_l}^{n} {n'}_{p_l > p_j}$$

We add to the permutations of A' the symbol x such as to retrieve the permutations of A. Suppose that p_l is at position i in a permutation of A', and p_l comes after x in the corresponding permutation of A, then we have one

permutation less in A with p_l at position i and one permutation more with p_l at position $i+1$. Hence, it follows that

$$\sum_{i=1}^{n+1} i \cdot n_i^{p_l} = \sum_{i=1}^{n} i \cdot n'^{p_l}_i + n_{p_l>x}.$$

Moreover, $n'_{p_l>p_j} = n_{p_l>p_j}$ for all $p_j \neq p_l$ and $p_j \neq x$. Hence,

$$\sum_{i=1}^{n+1} i \cdot n_i^{p_l} = \sum_{i=1}^{n} i \cdot n'^{p_l}_i + n_{p_l>x} = m + \sum_{j=1, p_j \neq p_l}^{n} n_{p_l>p_j} + n_{p_l>x}.$$

Since x and p_l are any two symbols from the $n+1$ symbols, expression (4.1) is valid for $n+1$. ∎

4.4 Real-world application

The environmental protection agency of Baden–Württemberg (Germany) has established a general monitoring system of bioindicators w.r.t. the accumulation of polluting chemical elements in different targets such as the herb layer, moss layer and leaves. In order to obtain a geographical overview [93], the state of Baden–Württemberg has been divided into 60 approximately homogeneous regions (Figure 4.2), and each region is represented by one measuring site where various chemical elements are determined as concentrations in the targets. As an illustration, we will use the pollution data of the herb layer in Table 4.3 to establish an objective ranking of the regions according to their pollution. Remark that for one region (region 55), no data is available, such that only 59 regions are to be ranked.

In order to rank the regions, we use three different approaches [40]. First, we compute the exact rank probabilities using the lattice of ideals representation (see Subsection 4.2.1) in order to obtain the average rank for each region. However, since memory requirements make this problem infeasible with present technology, we first apply a transformation on the data (cf. [21]). It might be reasonable to assume that for low concentrations of a polluting chemical element a modification of the data is allowed. For each

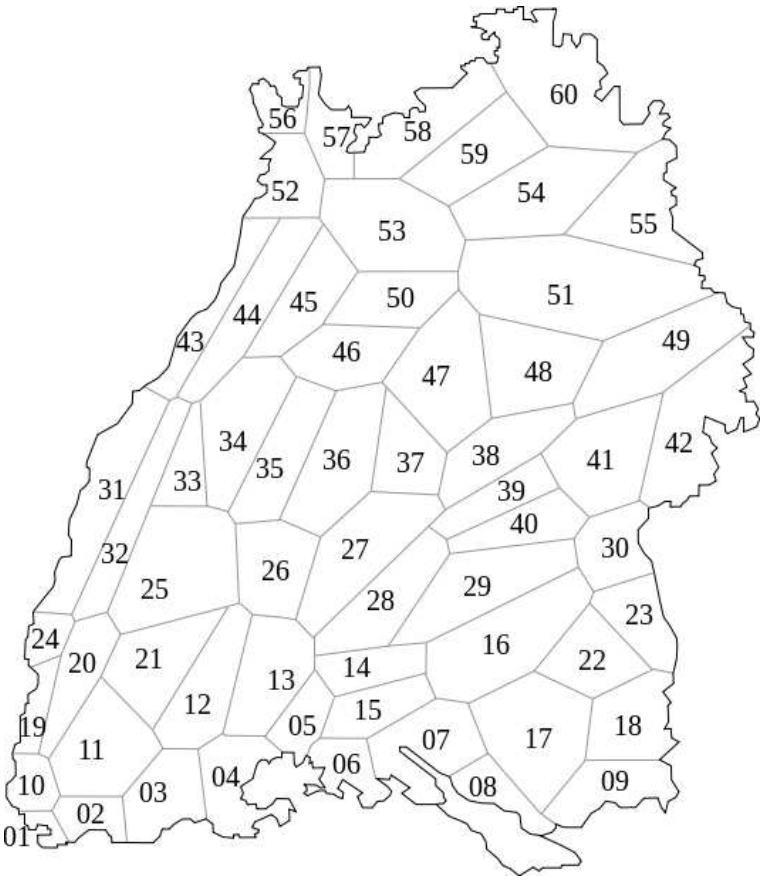


Figure 4.2: Division of Baden–Württemberg into 60 regions

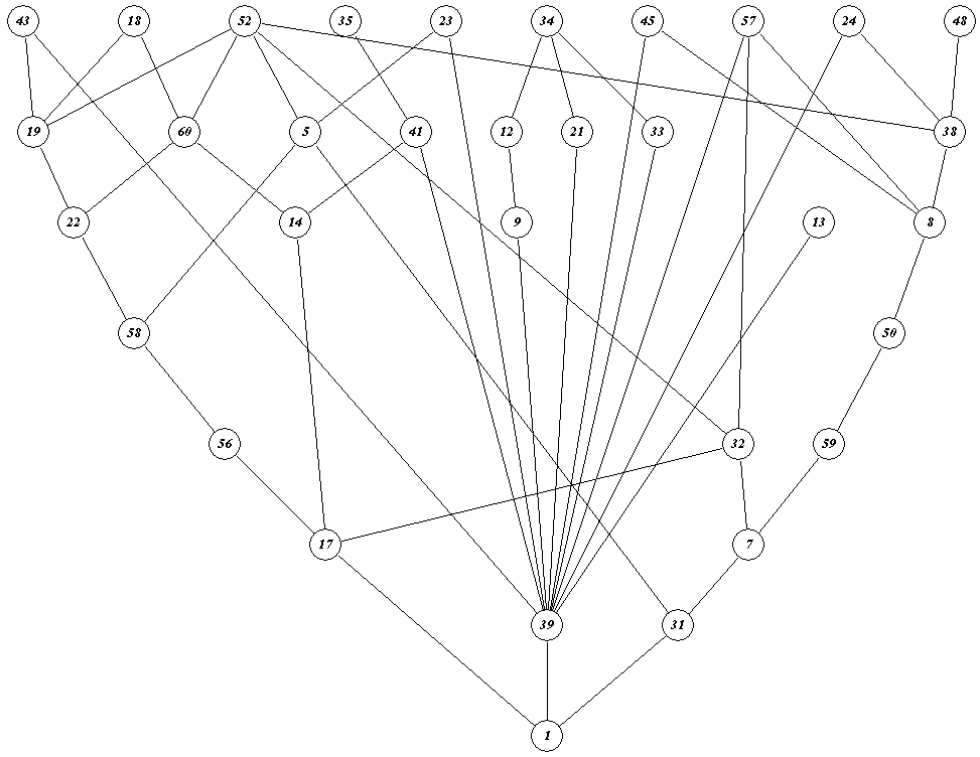


Figure 4.3: Hasse diagram corresponding to the reduced data set.

criterion c_i, a threshold s_i is chosen resulting in the modified criterion c_i'

$$c_i'(x) := \begin{cases} c_i(x) & \text{if} \quad c_i(x) \geq s_i \\ -\infty & \text{if} \quad c_i(x) < s_i \end{cases} .$$

Each threshold is chosen such that 25% of the objects retain their original value. The thresholds s_i are shown in Table 4.4, and correspond to criterion values $c_i(x_i)$ where x_i is the 44th object sorted increasingly according to criterion c_i. This transformation reduces the number of unique regions to 33 equivalence classes (see Table 4.5 for the non-trivial classes). The Hasse diagram corresponding to this reduced data set is shown in Figure 4.3, generated by means of the software tool WHASSE [19].

Subsequently, all rank probabilities for the 33 classes and the corresponding

average ranks are computed. As a side result the number of linear extensions and ideals is obtained. Since our poset has 15 909 106 995 077 310 937 728 linear extensions, *i.e.* possible rankings, the naive approach to enumerate all linear extensions in order to derive the rank probabilities is clearly infeasible in this case. A drawback of our approach is the huge amount of memory required to store the lattice of ideals which contains 108 862 ideals. However, this is clearly offset by the limited time needed to build the lattice of ideals and compute the rank probabilities: on a Linux-based 2.8Ghz machine with 2.5Gb of memory, our Java implementation only needs a couple of seconds. In order to verify the correctness of the implementations of the algorithms to compute the (mutual) rank probabilities, for posets on up to 9 elements all rank probability and mutual rank probability relations were compared with the results obtained by an independent approach based on the Varol-Rotem algorithm listed in Algorithm 3.1. For each poset, all linear extensions were enumerated and the rank probability relation and mutual rank probability relation were computed. Furthermore, the exact results for the data sets we consider in this chapter have been compared with the results obtained by the approximative algorithms based on the Markov chain Monte Carlo methods. In Table 4.6 the 10 regions with the highest average ranks ρ are shown, and Figure 4.4 shows a scatter plot of the average ranks of each region. Remark that non-representative regions are indicated with an X instead of a diamond and obtained the same average rank as their representative region.

In the last approach, we compute approximate average ranks of all 59 regions (see Figure 4.5). We used the same parameters as in the second approach. Although most regions having a high average rank in the transformed data set obtain a high approximate average rank in the original data set and vice versa, for some regions there is a striking difference. Region 34, for example, receives the second highest average rank in the first approach, while it drops to an average position in this approach. The reason for this discrepancy is that this region is characterized by an extremely low concentration of lead and sulfur, while very high concentrations of cadmium and zinc are measured. Due to the transformation, the impact of the high cadmium and zinc concentrations clearly intensifies, therefore favoring a higher average rank.

Finally, to conclude our real-world example, we also compute the exact mutual rank probabilities in the reduced data set. The results for the regions with the 10 highest average ranks, which turn out to be the set of max-

imal elements in the reduced data set, are shown in Table 4.7. Remark that the two regions with the highest average ranks, *i.e.* region 52 and 34, have mutual rank probabilities which are very close to $1/2$ (*i.e.* $0,52$ and $0,48$), indicating that they are rather indifferent. This observation is also confirmed by the fact that their average ranks are very close to each other. For regions 35 and 57, the mutual rank probabilities are even exactly $1/2$. To the contrary, the probability that region 52 is ranked higher than region 45 is $0,73$, indicating that region 52 is clearly to be preferred over region 45.

region	Pb	Cd	Zn	S	region	Pb	Cd	Zn	S
6	1,0	0,07	29	1750	36	1,2	0,05	31	1570
8	1,5	0,07	27	1750	46	0,8	0,09	33	1680
7	1,2	0,09	28	1600	50	1,4	0,13	29	1730
17	0,6	0,06	36	1820	53	1,0	0,12	36	1750
9	0,09	0,2	850	580	45	1,5	0,17	45	1780
16	1,0	0,12	32	1520	54	0,7	0,1	26	1750
22	1,0	0,03	28	2150	59	1,3	0,13	26	1470
18	0,5	0,43	28	4030	60	1,0	0,2	32	2160
30	0,8	0,08	27	1610	58	1,0	0,11	28	1980
23	1,1	0,04	42	2000	57	1,7	0,15	39	1850
15	0,9	0,1	24	1670	35	0,08	0,24	720	1960
14	1,0	0,17	34	1830	34	0,14	0,39	950	400
5	1,1	0,1	32	1990	33	0,16	0,26	800	530
28	0,9	0,05	34	1670	25	0,9	0,09	35	1460
39	1,0	0,1	38	1740	12	0,16	0,23	910	1460
40	0,7	0,06	34	1770	21	0,06	0,24	830	620
29	0,6	0,14	27	1680	11	0,9	0,08	27	1720
41	0,7	0,17	39	1840	2	0,7	0,14	27	1770
42	0,7	0,1	33	1690	1	1,0	0,04	21	1540
27	0,1	0,12	26	1600	10	1,0	0,03	29	1780
38	1,7	0,18	34	1720	20	1,5	0,14	32	1730
49	0,8	0,11	37	1680	24	1,7	0,18	39	1740
37	0,6	0,12	33	1580	31	1,1	0,15	28	1740
47	1,1	0,11	25	1650	32	1,2	0,03	35	1820
48	2,3	0,42	33	1600	19	0,8	0,01	18	4030
51	0,8	0,14	22	1640	43	0,5	0,11	39	4030
4	0,8	0,02	26	1790	44	0,8	0,08	38	1800
3	0,8	0,14	31	1710	52	2,0	0,23	36	4030
13	0,18	0,18	1160	350	56	1,0	0,11	34	1970
26	0,8	0,05	19	1620					

Table 4.3: Herb layer pollution with Pb, Cd, Zn and S of each region in Baden–Württemberg (in mg per kg dry weight).

c_i	s_i
Pb	1,1 mg/kg
Cd	0,17 mg/kg
Zn	38 mg/kg
S	1820 mg/kg

Table 4.4: The chosen threshold s_i (in mg per kg dry weight) for each attribute c_i.

regions	number	pattern
$\{\mathbf{1}, 2, 3, 4, 6, 10, 11, 15, 16, 25, 26, 27,$ $28, 29, 30, 37, 40, 42, 46, 49, 51, 53, 54\}$	23	$(-\infty; -\infty; -\infty; -\infty)$
$\{\mathbf{39}, 44\}$	2	$(-\infty; -\infty; 38; -\infty)$
$\{\mathbf{31}, 47\}$	2	$(1,1; -\infty; -\infty; -\infty)$
$\{\mathbf{7}, 36\}$	2	$(1,2; -\infty; -\infty; -\infty)$
$\{\mathbf{8}, 20\}$	2	$(1,5; -\infty; -\infty; -\infty)$

Table 4.5: Equivalence classes with more than one element and their representative element in boldface.

x	$\rho(x)$	$\hat{\rho}(x)$
52	29,21	29,20
34	29,12	29,10
18	27,91	27,91
24	26,94	26,93
48	26,93	26,92
43	26,37	26,37
57	26,27	26,28
35	26,04	26,06
23	25,51	25,51
45	24,86	24,88

Table 4.6: Regions with the 10 highest average ranks ρ and their approximations $\hat{\rho}$ by the second approach.

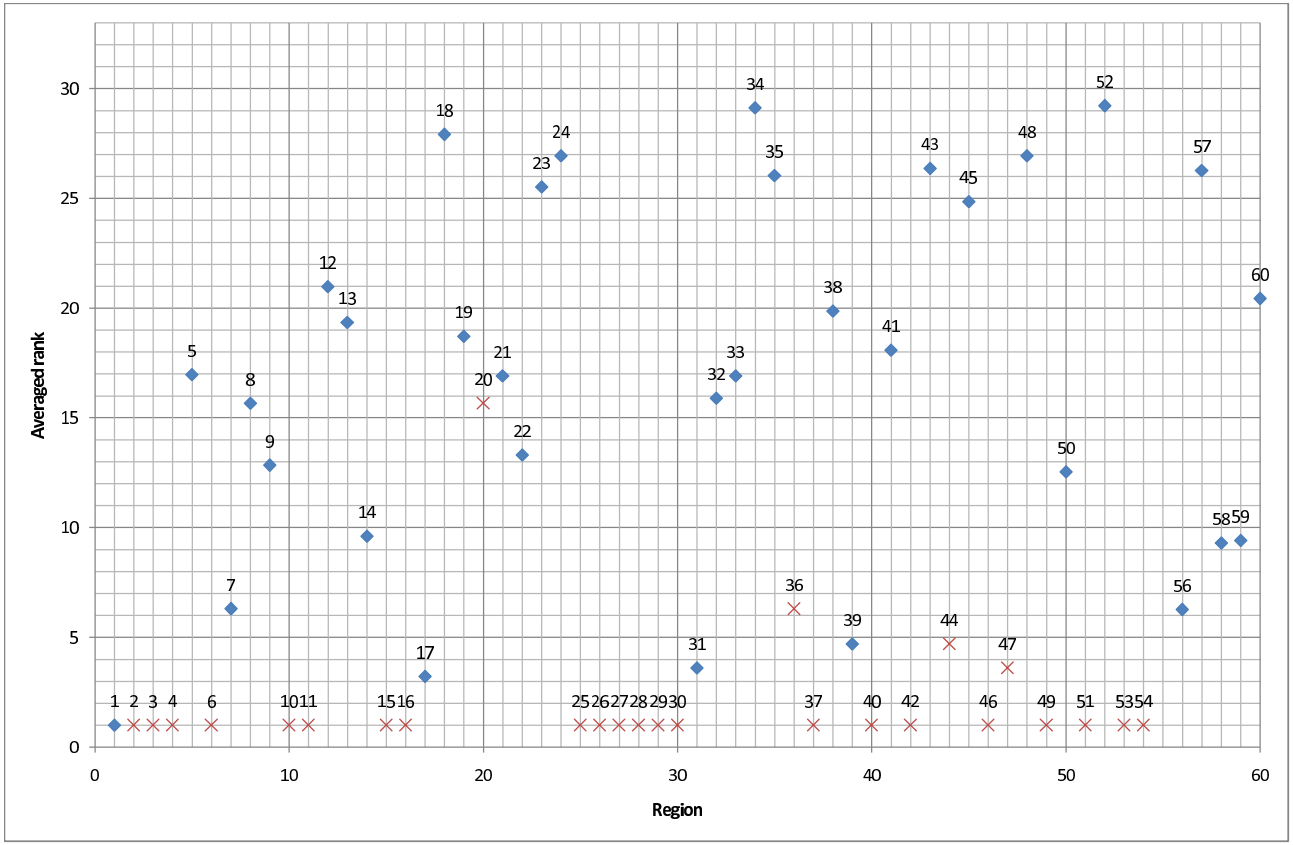


Figure 4.4: Herb layer pollution: the average rank for each region after the transformation is applied. Remark that all non-representative regions are indicated with an X instead of a diamond.

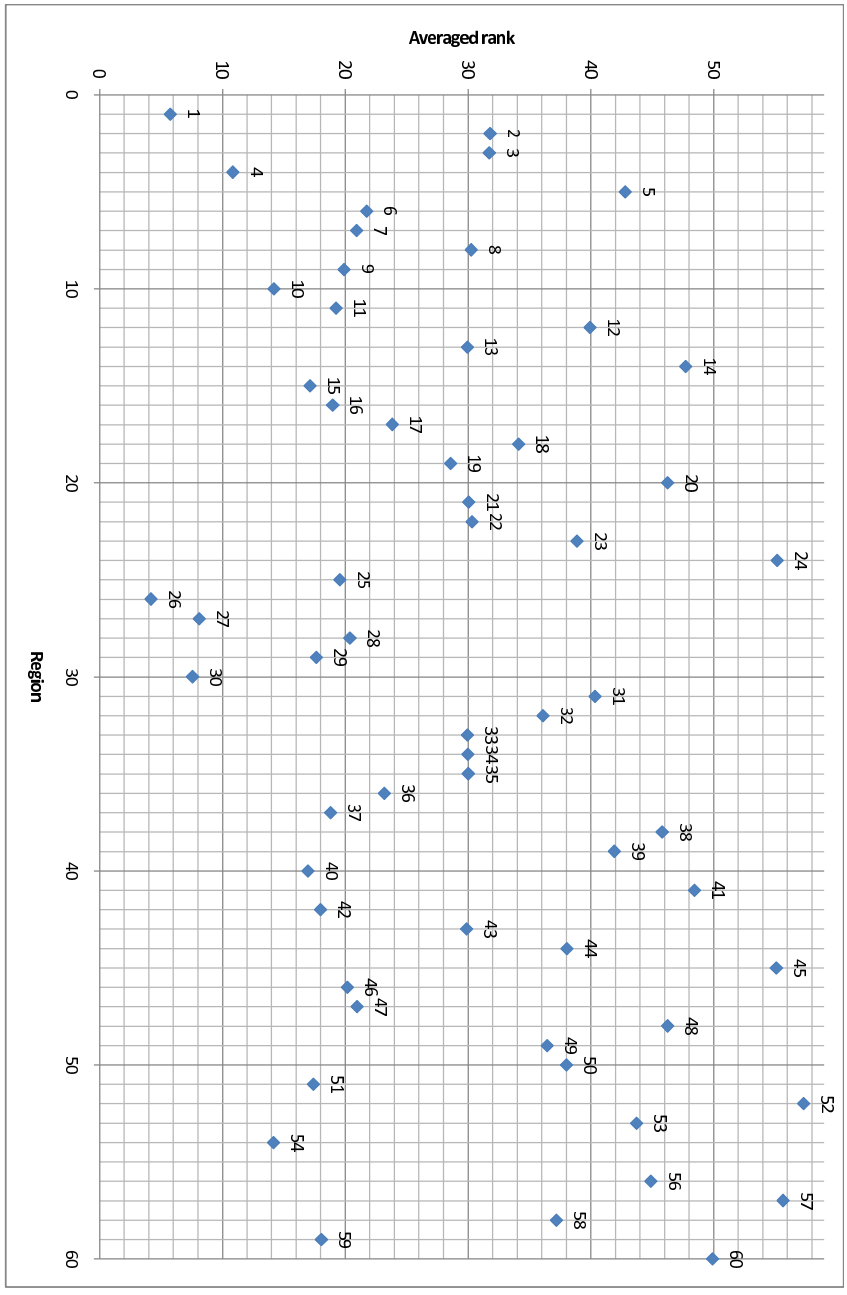


Figure 4.5: Herb layer pollution: an approximate average rank for each of the 59 regions.

$x \backslash y$	18	23	24	34	35	43	45	48	52	57
18	0	0,62	0,56	0,41	0,58	0,59	0,65	0,56	0,41	0,59
23	0,38	0	0,44	0,31	0,47	0,46	0,54	0,44	0,31	0,47
24	0,44	0,56	0	0,36	0,53	0,53	0,61	0,5	0,36	0,54
34	0,59	0,69	0,64	0	0,66	0,66	0,72	0,64	0,52	0,66
35	0,42	0,53	0,47	0,34	0	0,49	0,56	0,47	0,35	0,5
43	0,41	0,54	0,47	0,34	0,51	0	0,57	0,47	0,34	0,51
45	0,35	0,46	0,39	0,28	0,44	0,43	0	0,39	0,27	0,43
48	0,44	0,56	0,5	0,36	0,53	0,53	0,61	0	0,36	0,54
52	0,59	0,69	0,64	0,48	0,65	0,66	0,73	0,64	0	0,67
57	0,41	0,53	0,46	0,34	0,5	0,49	0,57	0,46	0,33	0

Table 4.7: Mutual rank probabilities Prob$(x > y)$ of the regions with the 10 highest average ranks in the reduced data set.

5 Transitivity of mutual rank probabilities

5.1 Introduction

5.1.1 Reciprocal relations

Definition 5.1.1. A *reciprocal relation* Q on a set of alternatives A is a mapping $Q : A^2 \to [0,1]$ such that for all $a, b \in A$, it holds that $Q(x,y) + Q(y,x) = 1$. ◇

A reciprocal relation [38, 48, 49] is sometimes also called a *probabilistic* or *ipsodual relation*.

Reciprocal relations serve as a popular representation of various preference models [33, 57, 113] where often the number $Q(x,y)$ expresses the *degree of preference* of alternative x over alternative y. If we confront a subject with two alternatives, say a and b, and ask which alternative is preferred, three answers are possible: a is preferred over b, b is preferred over a or a and b are equally preferred, *i.e.* the subject is *indifferent* with respect to a and b. Remark that although we allow for indifference, we assume that incomparability cannot occur. Note that incomparability here has to be seen in the context of preference structures rather than in the context of posets. To obtain a degree to which a is preferred over b, the proportion of answers

in which the subject prefers a with respect to a total of n answers can be used:

$$P(a,b) = \frac{\#\{a \text{ is preferred to } b\}}{n}.$$

As we allow for indifference, the relation P is not necessarily a reciprocal relation. A reciprocal relation Q can, however, be easily constructed from P:

$$Q(a,b) = P(a,b) + \frac{1}{2}I(a,b),$$

where

$$I(a,b) = \frac{\#\{a \text{ and } b \text{ are equally preferred}\}}{n}.$$

It is easily verified that the relation Q is a reciprocal relation. Note that a subject need not necessarily be asked her preference multiple times: an estimate of the degree of preference could be obtained otherwise.

Reciprocal relations are a convenient tool for expressing the result of a pairwise comparison of a set of alternatives [33] and appear in various fields such as game theory [51], voting theory [65, 94] and psychological studies on preference and discrimination in decision-making methods [50]. In fuzzy set theory, reciprocal relations are used for representing intensities of preference [29, 75, 115]; in group decision making, they represent collective preferences [64, 75], and in social choice theory, given preferences are expressed in terms of reciprocal relations [31, 51, 74, 82, 94].

5.1.2 Aggregation operators

Transitivity properties are described by using specific classes of *aggregation operators*, some of which are introduced below.

Definition 5.1.2. A binary operation $T : [0,1]^2 \to [0,1]$ is called a *t-norm* has the following properties:

(i) 1 is the neutral element: $T(x,1) = x = T(1,x)$, $\forall x \in [0,1]$

(ii) monotonicity: T is increasing in each variable

(iii) commutatitivity: $T(x,y) = T(y,x)$, $\forall x,y \in [0,1]$

(iv) associativity: $T(x,T(y,z)) = T(T(x,y),z)$, $\forall x,y,z \in [0,1]$

◇

A related concept is that of a *t-conorm*, which is a binary operation on $[0,1]$ satisfying conditions (ii)-(iv) and which has 0 as the neutral element. Furthermore, to any norm T there corresponds a dual conorm S defined by

$$S(x,y) = 1 - T(1-x, 1-y). \tag{5.1}$$

The smallest t-norm is the *drastic product* $T_{\mathbf{D}}$, which is right-continuous only and is 0 everywhere up to the boundary condition $T_{\mathbf{D}}(x,1) = T_{\mathbf{D}}(1,x) = x$. The three main continuous t-norms are the *minimum operator* $T_{\mathbf{M}}$, *i.e.* $T_{\mathbf{M}}(x,y) = \min(x,y)$, the *algebraic product* $T_{\mathbf{P}}$, *i.e.* $T_{\mathbf{P}}(x,y) = x \cdot y$, and the *Łukasiewicz t-norm* $T_{\mathbf{L}}(x,y) = \max(0, x+y-1)$. It holds that $T_{\mathbf{D}} \leq T_{\mathbf{L}} \leq T_{\mathbf{P}} \leq T_{\mathbf{M}}$, according to the usual ordering of functions

Remark 5.1.3. The three continuous t-norms $T_{\mathbf{M}}$, $T_{\mathbf{P}}$ and $T_{\mathbf{L}}$ belong to an important parametric family of t-norms, namely the *Frank family* $(T_\lambda^{\mathbf{F}})_{\lambda\in[0,\infty]}$. For $\lambda \in]0,1[\cup]1,\infty[$, the t-norm $T_\lambda^{\mathbf{F}}$ is defined by

$$T_\lambda^{\mathbf{F}}(x,y) = \log_\lambda\left(1 + \frac{(\lambda^x - 1)(\lambda^y - 1)}{\lambda - 1}\right).$$

As limit cases, one obtains $T_{\mathbf{M}}(\lambda \to 0)$, $T_{\mathbf{P}}(\lambda \to 1)$ and $T_{\mathbf{L}}(\lambda \to \infty)$.

Definition 5.1.4. A binary operation $C : [0,1]^2 \to [0,1]$ is called a *quasi-copula* if it has the following properties:

(i) 1 is the neutral element: $C(x,1) = x = C(1,x)$, $\forall x \in [0,1]$

(i') absorbing element: $C(x,0) = C(0,x) = 0$, $\forall x \in [0,1]$

(ii) monotonicity: C is increasing in each variable

(iii) 1-Lipschitz property: $|C(x_1,y_1) - C(x_2,y_2)| \leq |x_1 - x_2| + |y_1 - y_2|$, $\forall x_1, x_2, y_1, y_2 \in [0,1]$

Moreover, the binary operation C is called a *copula* if, instead of the 1-Lipschitz property, the following stronger property holds:

(iv) moderate growth (2-increasing):
$(x_1 \leq x_2 \wedge y_1 \leq y_2) \Rightarrow C(x_1,y_1) + C(x_2,y_2) \geq C(x_1,y_2) + C(x_2,y_1)$, $\forall x_1, x_2, y_1, y_2 \in [0,1]$

Note that condition (i') follows from conditions (i) and (ii). For a copula, condition (ii) can be omitted as it follows from (iv) and (i'). Remark that any copula is a quasi-copula, and therefore has the 1-Lipschitz property; the opposite, however, is not true. ◇

Note that 1-Lipschitz continuous t-norms are nothing else but associative copulas. Note that $T_{\mathbf{M}}$ is the greatest t-norm and copula, $T_{\mathbf{L}}$ is the smallest copula, while $T_{\mathbf{P}}$ is the independence copula.

5.1.3 Transitivity

Transitivity is a simple, yet powerful property of relations.

Definition 5.1.5. A binary relation R on a universe A is called *transitive* if for any $a, b, c \in A$ it holds that

$$((a,b) \in R \wedge (b,c) \in R) \;\Rightarrow\; (a,c) \in R. \tag{5.2}$$

◇

Example 5.1. The relation "*is a subset of*" for a set of sets is an example of a transitive relation: if for three sets A, B and C it holds that *A is a subset of B* and *B is a subset of C* then necessarily *A is a subset of C*. •

Identifying a relation with its characteristic mapping, *i.e.* defining

$$R(a,b) = \begin{cases} 1 & \text{if } (a,b) \in R \\ 0 & \text{if } (a,b) \notin R \end{cases}$$

transitivity can be stated equivalently as

$$(R(a,b) = 1 \wedge R(b,c) = 1) \;\Rightarrow\; R(a,c) = 1.$$

However, many other equivalent formulations may be devised, such as

$$(R(a,b) \geq \alpha \wedge R(b,c) \geq \alpha) \;\Rightarrow\; R(a,c) \geq \alpha, \tag{5.3}$$

for any $\alpha > 0$. Alternatively, transitivity can also be expressed in the following functional form:

$$\min(R(a,b), R(b,c)) \leq R(a,c). \tag{5.4}$$

Note that on $\{0,1\}^2$ the minimum operator is the Boolean conjunction: when one or more of the arguments is 0, the result is 0.

5.1.3.1 Reciprocal relations

Transitivity properties of reciprocal relations are usually of the form in (5.3). Various kinds of stochastic transitivity for reciprocal relations exist and will be discussed in Section 5.3.

We remark that reciprocity is linked with completeness. Let R be a complete $\{0,1\}$-valued relation on A, *i.e.* $\max(R(a,b),R(b,a)) = 1$ for any $a,b \in A$, then R has an equivalent $\{0,\frac{1}{2},1\}$-valued reciprocal representation Q given by

$$Q(a,b) = \begin{cases} 1 & \text{if } R(a,b) = 1 \text{ and } R(b,a) = 0, \\ \frac{1}{2} & \text{if } R(a,b) = R(b,a) = 1, \\ 0 & \text{if } R(a,b) = 0 \text{ and } R(b,a) = 1, \end{cases} \tag{5.5}$$

which can be easily rewritten in the more compact form

$$Q(a,b) = \frac{1 + R(a,b) - R(b,a)}{2}. \tag{5.6}$$

Remark that a complete relation that is transitive in the sense of (5.2) is a weak order relation.

One easily verifies that R is transitive if and only if its reciprocal representation Q satisfies, for any $a,b,c \in A$:

$$(Q(a,b) \geq 1/2 \wedge Q(b,c) \geq 1/2) \;\Rightarrow\; Q(a,c) = \max(Q(a,b),Q(b,c)) \tag{5.7}$$

5.1.3.2 Fuzzy relations

A *fuzzy relation* R on A is a $A^2 \to [0,1]$ mapping expressing the degree of relationship between elements of A. The extreme case where $R(a,b) = 0$ means that a and b are not related at all, while $R(a,b) = 1$ expresses full relationship. A value $R(a,b) \in]0,1[$ expresses partial relationship.

In the setting of fuzzy set theory, a T-norm is used as a generalization for the Boolean conjunction in formulation (5.4).

Definition 5.1.6. Let T be a t-norm. A fuzzy relation R on A is called *T-transitive* if for any $a,b,c \in A$ it holds that

$$T(R(a,b),R(b,c)) \leq R(a,c). \tag{5.8}$$

◇

By using a *fuzzy preference relation* R, a fuzzy model of preferences could be used to obtain a reciprocal relation. In the specific case of a fuzzy preference relation, $R(a,b)$ expresses the (subjective) judgement of degree in which a is preferred to b. When $R(a,b) = 0$, b is completely preferred to a, while $R(a,b) = 1$ indicates that a is completely preferred to b.

Remark 5.1.7. Note that although the condition $R(a,b) + R(b,a) = 1$ is in general not required for fuzzy preference relations, we will restrict ourselves to fuzzy preference relations that obey this condition, and thus are reciprocal. Although formally reciprocal relations can be seen as a special kind of fuzzy relations, they are not equipped with the same semantics.

Definition 5.1.8. A binary operation $f : [0,1]^2 \to [0,1]$ is called a *conjunctor* if it has the following properties:

(i) its restriction to $\{0,1\}$ coincides with the Boolean conjunction;

(ii) monotonicity: f is increasing in each variable.

◇

Definition 5.1.9. Let f be a conjunctor. A reciprocal relation Q on A is called *f-transitive* if for any $a,b,c \in A$ it holds that

$$f(Q(a,b),Q(b,c)) \leq Q(a,c). \tag{5.9}$$

◇

We have already seen two particular classes of conjunctors with neutral element 1: the class of t-norms and the class of (quasi)-copulas, which, just as t-norms, finds its origin in the study of probabilistic metric spaces.

For two conjunctors $f_1 \leq f_2$, it clearly holds that f_1-transitivity implies f_2-transitivity. However, it is important to note that $f_1 \leq f_2$ is not a necessary condition for the latter implication to hold for reciprocal relations.

5.2 Cycle-transitivity

The framework of cycle-transitivity [34, 38] presented in this section unifies reciprocal transitivity and fuzzy transitivity, which are basically two different concepts. It is, however, useful to have the class of transitivity properties as broad as possible, since the standard types of stochastic transitivity are insufficient to model e.g. human preferences described by reciprocal relations. In experimental studies, it has been observed that when persons are asked to express their preference between the elements of some set pairwisely, it is not uncommon that cycles occur in the resulting preference relation. Moreover, it will turn out that the mutual rank probability relation M_P studied in this chapter can also contain cycles. The framework of cycle-transitivity offers a convenient way out, since it does not exclude the possibility of this cyclic behavior. Although T-transitivity has been devised for fuzzy relations, which are not necessarily reciprocal relations, we will start by studying $T_{\mathbf{P}}$-transitivity for reciprocal relations. The observations made will form a motivation for the introduction of the concept of cycle-transitivity.

Consider an arbitrary set A of alternatives. For a reciprocal relation Q on A, we write $q_{ab} := Q(a,b)$. For any $a,b,c \in A$ let

$$\begin{aligned} \alpha_{abc} &= \min(q_{ab}, q_{bc}, q_{ca}), \\ \beta_{abc} &= \operatorname{median}(q_{ab}, q_{bc}, q_{ca}), \\ \gamma_{abc} &= \max(q_{ab}, q_{bc}, q_{ca}). \end{aligned}$$

Note that it is not necessary for A to be countable. It now obviously holds that

$$\alpha_{abc} \leq \beta_{abc} \leq \gamma_{abc}, \tag{5.10}$$

and also

$$\alpha_{abc} = \alpha_{bca} = \alpha_{cab}, \quad \beta_{abc} = \beta_{bca} = \beta_{cab}, \quad \gamma_{abc} = \gamma_{bca} = \gamma_{cab}. \tag{5.11}$$

Due to the reciprocal nature of Q, it also holds that

$$\alpha_{cba} = 1 - \gamma_{abc}, \quad \beta_{cba} = 1 - \beta_{abc}, \quad \gamma_{cba} = 1 - \alpha_{abc}. \tag{5.12}$$

5.2.1 $T_{\mathbf{P}}$-transitivity for reciprocal relations

Let us consider $T_{\mathbf{P}}$-transitivity for a reciprocal relation Q on A. For any $a, b, c \in A$, six conditions are to be satisfied:

$$q_{ac}\, q_{cb} \leq q_{ab}, \quad q_{ba}\, q_{ac} \leq q_{bc}, \quad q_{cb}\, q_{ba} \leq q_{ca},$$
$$q_{bc}\, q_{ca} \leq q_{ba}, \quad q_{ca}\, q_{ab} \leq q_{cb}, \quad q_{ab}\, q_{bc} \leq q_{ac}.$$

Since Q is reciprocal, these conditions can be expressed in terms of α_{abc}, β_{abc} and γ_{abc} solely, as follows

$$\begin{aligned}
&(1-\beta_{abc})(1-\gamma_{abc}) \leq \alpha_{abc}, \quad \beta_{abc}\,\gamma_{abc} \leq 1-\alpha_{abc},\\
&(1-\alpha_{abc})(1-\gamma_{abc}) \leq \beta_{abc}, \quad \alpha_{abc}\,\gamma_{abc} \leq 1-\beta_{abc},\\
&(1-\alpha_{abc})(1-\beta_{abc}) \leq \gamma_{abc}, \quad \alpha_{abc}\,\beta_{abc} \leq 1-\gamma_{abc}.
\end{aligned} \tag{5.13}$$

The three left-hand inequalities of (5.13) can be rewritten as

$$\begin{aligned}
\beta_{abc}\,\gamma_{abc} &\leq \alpha_{abc}+\beta_{abc}+\gamma_{abc}-1,\\
\alpha_{abc}\,\gamma_{abc} &\leq \alpha_{abc}+\beta_{abc}+\gamma_{abc}-1,\\
\alpha_{abc}\,\beta_{abc} &\leq \alpha_{abc}+\beta_{abc}+\gamma_{abc}-1.
\end{aligned}$$

From (5.10) it follows that $\alpha_{abc}\,\beta_{abc} \leq \alpha_{abc}\,\gamma_{abc} \leq \beta_{abc}\gamma_{abc}$. Therefore, only the first inequality should be withheld as a condition for $T_{\mathbf{P}}$-transitivity.

Similarly, the three right-hand inequalities of (5.13) can be rewritten as

$$\begin{aligned}
\alpha_{abc}+\beta_{abc}+\gamma_{abc}-1 &\leq 1-(1-\beta_{abc})(1-\gamma_{abc}),\\
\alpha_{abc}+\beta_{abc}+\gamma_{abc}-1 &\leq 1-(1-\alpha_{abc})(1-\gamma_{abc}),\\
\alpha_{abc}+\beta_{abc}+\gamma_{abc}-1 &\leq 1-(1-\alpha_{abc})(1-\beta_{abc}).
\end{aligned}$$

From (5.10) it now follows that only the last inequality should be retained. The six inequalities in (5.13) are therefore equivalent to the double inequality

$$\beta_{abc}\gamma_{abc} \leq \alpha_{abc}+\beta_{abc}+\gamma_{abc}-1 \leq 1-(1-\alpha_{abc})(1-\beta_{abc}). \tag{5.14}$$

The way we arrived at this double inequality immediately shows that if it holds for $a, b, c \in A$, then it also holds for all permutations of (a, b, c).

If we denote the upper and lower bounds of (5.14) as $u(\alpha_{abc}, \beta_{abc})$ and $l(\beta_{abc}, \gamma_{abc})$, respectively, we observe the following type of duality:

$$l(x, y) = 1 - u(1-y, 1-x), \ \forall\, 0 \leq x \leq y \leq 1. \tag{5.15}$$

5.2.2 Definition

The simple formulation (5.14)-(5.15) of $T_{\mathbf{P}}$-transitivity for reciprocal relations has been the source of inspiration for a new way of describing the transitivity of probabilistic relations. Let us denote $\Delta = \{(x,y,z) \in [0,1]^3 \mid x \leq y \leq z\}$ and consider a function $U : \Delta \to \mathbb{R}$, then, in analogy to (5.14), we could call a probabilistic relation Q on A transitive with respect to U if for any $a,b,c \in A$ it holds that

$$\alpha_{abc} + \beta_{abc} + \gamma_{abc} - 1 \leq U(\alpha_{abc}, \beta_{abc}\gamma_{abc}), \tag{5.16}$$

and that

$$1 - U(1-\gamma_{abc}, 1-\beta_{abc}, 1-\alpha_{abc}) \leq \alpha_{abc} + \beta_{abc} + \gamma_{abc} - 1. \tag{5.17}$$

Remark that in the case of $T_{\mathbf{P}}$-transitivity, the corresponding function $U_{\mathbf{P}}$ is given by

$$U_{\mathbf{P}}(\alpha,\beta,\gamma) = 1 - (1-\alpha)(1-\beta) = \alpha + \beta - \alpha\,\beta. \tag{5.18}$$

The minimal requirement we will impose is that the reciprocal representation Q of any transitive complete relation R given in (5.6) satisfies any form of cycle-transitivity. Therefore, due to the condition (5.7) for R being transitive, we know that $(\alpha_{abc}, \beta_{abc}, \gamma_{abc})$ is either $(0,0,1)$, $(0,1/2,1)$, $(1/2,1/2,1/2)$ or $(0,1,1)$. Insisting that (5.16)-(5.17) is valid for these possible values for $(\alpha_{abc}, \beta_{abc}, \gamma_{abc})$ yields the following conditions on U:

$$\begin{aligned} U(0,1/2,1) \geq 1/2, \quad U(1/2,1/2,1/2) \geq 1/2, \\ U(0,0,1) \geq 0, \qquad U(0,1,1) \geq 1. \end{aligned} \tag{5.19}$$

Remark 5.2.1. Similarly, we could insist that the only $\{0,1/2,1\}$-valued reciprocal relations that are transitive with respect to U are the reciprocal representations of transitive complete relations. As this requirement would limit the generality of the framework considerable, it is not imposed.

Definition 5.2.2. A function $U : \Delta \to \mathbb{R}$ is called an *upper bound function* if it satisfies:

(i) $U(0,0,1) \geq 0$ and $(0,1,1) \geq 1$

(ii) for any $(\alpha,\beta,\gamma) \in \Delta$:

$$U(\alpha,\beta,\gamma) + U(1-\gamma, 1-\beta, 1-\alpha) \geq 1 \tag{5.20}$$

◇

Note that the conditions $U(0,1/2,1) \geq 1/2$ and $U(1/2,1/2,1/2) \geq 1/2$ follow immediately from (5.20) when $(\alpha,\beta,\gamma) = (0,1/2,1)$ and $(\alpha,\beta,\gamma) = (1/2,1/2,1/2)$ respectively.

The function $L : \Delta \to \mathbb{R}$ defined by

$$L(\alpha,\beta,\gamma) = 1 - U(1-\gamma, 1-\beta, 1-\alpha) \tag{5.21}$$

is called the *dual lower bound function* of a given upper bound function U. Inequality (5.20) then simply expresses that $L \leq U$. Remark furthermore that there is no monotonicity constraint for U.

Definition 5.2.3. A reciprocal relation Q on A is called *cycle-transitive* with respect to an upper bound function U if for any $a,b,c \in A$ it holds that

$$L(\alpha_{abc},\beta_{abc},\gamma_{abc}) \leq \alpha_{abc} + \beta_{abc} + \gamma_{abc} - 1 \leq U(\alpha_{abc},\beta_{abc},\gamma_{abc}), \tag{5.22}$$

where L is the dual lower bound function of U. ◇

Remark that due to (5.11) all cyclic permutations of (a,b,c) yield conditions equivalent to (5.22). Let us now take a permutation of $(a,b,c) \in A^3$ that is not a cyclic permutation, e.g. (c,b,a), such that the following condition is obtained:

$$L(\alpha_{cba},\beta_{cba},\gamma_{cba}) \leq \alpha_{cba} + \beta_{cba} + \gamma_{cba} - 1 \leq U(\alpha_{cba},\beta_{cba},\gamma_{cba}),$$

which is, due to (5.12), equivalent to

$$\begin{aligned} 1 - L(1-\gamma_{abc}, 1-\beta_{abc}, 1-\alpha_{abc}) &\geq \alpha_{abc} + \beta_{abc} + \gamma_{abc} - 1 \\ &\geq 1 - U(1-\gamma_{abc}, 1-\beta_{abc}, 1-\alpha_{abc}). \end{aligned}$$

Using the built-in duality of U and L it can easily be shown this condition is equivalent to (5.22). Analogous observations for the cyclic permutations (b,a,c) and (a,c,b) can be made. In practice, therefore, it suffices to verify whether (5.22) holds for a single permutation of $(a,b,c) \in A^3$.

Alternatively, it is also sufficient to verify the right-hand inequality (or equivalently, the left-hand inequality) for two permutations of any $(a,b,c) \in A^3$ that aren't cyclic permutations of one another, e.g. (a,b,c) and (c,b,a). Indeed, consider the condition corresponding to the permutation (c,b,a):

$$\alpha_{cba} + \beta_{cba} + \gamma_{cba} - 1 \quad \leq U(\alpha_{cba},\beta_{cba},\gamma_{cba}).$$

Due to (5.12), it can be rewritten as

$$(1-\gamma_{abc})+(1-\beta_{abc})+(1-\alpha_{abc})-1 \leq U(1-\gamma_{abc}, 1-\beta_{abc}, 1-\alpha_{abc}),$$

which is, following directly from the definition of the dual lower bound, equivalent to

$$\alpha_{abc}+\beta_{abc}+\gamma_{abc}-1 \geq L(\alpha_{abc}, \beta_{abc}, \gamma_{abc}).$$

Together with the condition

$$\alpha_{abc}+\beta_{abc}+\gamma_{abc}-1 \leq U(\alpha_{abc}, \beta_{abc}, \gamma_{abc}),$$

we obtain the definition (5.22) of cycle-transitivity.

Remark 5.2.4. If the equality in (5.20) holds for all $(\alpha, \beta, \gamma) \in \Delta$, *i.e.*

$$U(\alpha, \beta, \gamma) + U(1-\gamma, 1-\beta, 1-\alpha) = 1, \tag{5.23}$$

then the upper bound function U is said to be *self-dual*, since in that case it coincides with its dual lower bound function L. The simplest self-dual upper bound function is given by the median, *i.e.* $U_{\mathbf{M}}(\alpha, \beta, \gamma) = \beta$.

Note that a value of $U(\alpha, \beta, \gamma)$ equal to 2 will often be used to express that for the given arguments there is no restriction at all. Indeed, $\alpha+\beta+\gamma-1$ is always upper bounded by 2.

For two upper bound functions $U_1 \leq U_2$, it clearly holds that cycle-transitivity with respect to U_1 implies cycle-transitivity with respect to U_2. However, it is clear that $U_1 \leq U_2$ is not a necessary condition for the latter implication to hold. Two upper bound functions U_1 and U_2 will be called *equivalent* if for any $(\alpha, \beta, \gamma) \in \Delta_3$ it holds that $\alpha+\beta+\gamma-1 \leq U_1(\alpha, \beta, \gamma)$ is equivalent to $\alpha+\beta+\gamma-1 \leq U_2(\alpha, \beta, \gamma)$. For instance, suppose that the inequality $\alpha+\beta+\gamma-1 \leq U_1(\alpha, \beta, \gamma)$ can be rewritten as $\alpha \leq h(\beta, \gamma)$, then an equivalent upper bound function U_2 is given by $U_2(\alpha, \beta, \gamma) = \beta+\gamma-1+h(\beta, \gamma)$. In this way, it is often possible to find an equivalent upper bound function in only two of the three variables α, β and γ.

5.3 A hierarchy of types of transitivity for reciprocal relations

Most of the types of transitivity encountered in applications involving reciprocal relations fit into the cycle-transitivity framework with upper bound functions that are polynomials of low degree.

The following propositions will be invoked multiple times throughout this chapter [38].

Proposition 5.3.1. *Consider a commutative conjunctor $f \leq T_{\mathbf{M}}$, then f-transitivity of reciprocal relations is equivalent to cycle-transitivity with respect to the upper bound U_f defined by*

$$U_f(\alpha,\beta,\gamma) = \min(\alpha+\beta-f(\alpha,\beta), \beta+\gamma-f(\beta,\gamma), \gamma+\alpha-f(\gamma,\alpha)) \,. \tag{5.24}$$

Moreover, if the conjunctor f satisfies some additional conditions, the upper bound function U_f is further simplified [38]. An important case concerns t-norms [79].

Proposition 5.3.2. *Consider a 1-Lipschitz continuous t-norm T, then T-transitivity of reciprocal relations is equivalent to cycle-transitivity with respect to the upper bound function U_T defined by*

$$U_T(\alpha,\beta,\gamma) = \alpha+\beta-T(\alpha,\beta) \,. \tag{5.25}$$

5.3.1 Constant upper bound functions

If the upper bound function U is a constant function, we distinguish the following interesting cases:

(i) $U(\alpha,\beta,\gamma) = 2$.
Cycle-transitivity with respect to this upper bound function imposes no restrictions and is trivially satisfied by any reciprocal relation. One could even argue that this is not a type of transitivity.

(ii) $U(\alpha, \beta, \gamma) = 1$.
It can be shown that an equivalent upper bound function is given by $U_L(\alpha, \beta, \gamma) = \min(\alpha + \beta, 1) = \alpha + \beta - \max(\alpha + \beta - 1, 0)$. In view of Proposition 5.3.2, this type of cycle-transitivity corresponds to $T_{\mathbf{L}}$-transitivity. Interestingly, $T_{\mathbf{L}}$-transitivity is also equivalent to the *triangle inequality*: a reciprocal relation Q on A is $T_{\mathbf{L}}$-transitive if and only if $Q(a, b) + Q(b, c) \leq Q(a, c)$, for any $(a, b, c) \in A^3$ [118].

Note that the function $U(\alpha, \beta, \gamma) = 1/2$ is not suitable as an upper bound function as the condition $U(0, 1, 1) \geq 1$ is not satisfied. Reciprocal relations satisfying the condition $\alpha_{abc} + \beta_{abc} + \gamma_{abc} = 3/2$, *i.e.* relations being cycle-transitive with respect to this specific U, are called (additive) consistent in literature [72]. Although it is often presented as a type of transitivity, it does not deserve to be called so, as it is even in general not satisfied by the reciprocal 3-valued representation of a transitive complete $\{0, 1\}$-valued relation.

5.3.2 Linear upper bound functions

If the upper bound function is a linear function, two cases are of particular interest:

(i) $U_M(\alpha, \beta, \gamma) = \beta$.
In view of Proposition 5.3.2, cycle-transitivity with respect to this upper bound function corresponds to $T_{\mathbf{M}}$-transitivity. One easily verifies that U_M is self-dual.

(ii) $U_p(\alpha, \beta, \gamma) = \gamma$.
Cycle-transitivity with respect to this upper bound function is equivalent to a type of transitivity known as *partial stochastic transitivity* [48]. A reciprocal relation Q on A is called partially stochastic transitive if for any $(a, b, c) \in A^3$ it holds that

$$(Q(a, b) > 1/2 \wedge Q(b, c) > 1/2) \;\Rightarrow\; Q(a, c) \geq \min(Q(a, b), Q(b, c)) \,. \tag{5.26}$$

We also retrieve interesting types of transitivity by considering piecewise linear upper bound functions, in particular, upper bound functions of the

form

$$U_g(\alpha,\beta,\gamma) = \begin{cases} \beta+\gamma-g(\beta,\gamma) & \text{if } \beta \geq 1/2 \wedge \alpha < 1/2\,, \\ 1/2 & \text{if } \alpha \geq 1/2\,, \\ 2 & \text{if } \beta < 1/2\,, \end{cases}$$

where $g : [1/2,1]^2 \to [0,1]$ is an increasing linear function satisfying

$$g(1/2,x) \leq x \text{ for any } x \in [1/2,1].$$

Cycle-transitivity with respect to the upper bound function U_g is equivalent to g-stochastic transitivity [38]. A reciprocal relation Q on A is called g-stochastic transitive if for any $(a,b,c) \in A^3$ it holds that

$$(Q(a,b) \geq 1/2 \wedge Q(b,c) \geq 1/2) \;\Rightarrow\; Q(a,c) \geq g(Q(a,b),Q(b,c))\,. \quad (5.27)$$

For some functions g, special names are given to the corresponding type of transitivity [57]:

(i) *strong stochastic transitivity*: $g(\beta,\gamma) = \max(\beta,\gamma) = \gamma$,

(ii) *moderate stochastic transitivity*: $g(\beta,\gamma) = \min(\beta,\gamma) = \beta$,

(iii) *weak stochastic transitivity*: $g(\beta,\gamma) = 1/2$.

5.3.3 Quadratic upper bound functions

An interesting quadratic upper bound function is the one associated with $T_{\mathbf{P}}$-transitivity. Inspired by this example, we distinguish the following three upper bound functions:

(i) *strong product transitivity*: $U_{\mathbf{P}}(\alpha,\beta,\gamma) = \alpha+\beta-\alpha\beta$.
Obviously, it is the cycle-transitive formulation of $T_{\mathbf{P}}$-transitivity.

(ii) *moderate product transitivity*: $U_{mp}(\alpha,\beta,\gamma) = \alpha+\gamma-\alpha\gamma$.
Cycle-transitivity with respect to this upper bound function has, to our knowledge, not yet been reported in literature. It will play an important role further in this chapter.

(iii) *weak product transitivity*: $U_D(\alpha,\beta,\gamma) = \beta+\gamma-\beta\gamma$.
Cycle-transitivity with respect to this upper bound function is called *dice transitivity* [46, 49].

5.3.4 Cubic upper bound functions

One cubic self-dual upper bound function is of particular interest, namely

$$U_m(\alpha, \beta, \gamma) = \alpha\beta + \alpha\gamma + \beta\gamma - 2\alpha\beta\gamma\,.$$

Cycle-transitivity with respect to the upper bound function U_m is equivalent to a type of transitivity called *multiplicative transitivity* [38, 115]. A reciprocal relation Q on A is called multiplicatively transitive if for any $(a, b, c) \in A^3$ it holds that

$$Q(a,b)Q(b,c)Q(c,a) = Q(b,a)Q(a,c)Q(c,b).$$

5.3.5 A taxonomy of types of transitivity

When considering different types of cycle-transitivity, we can try to distinguish weaker or stronger types. Obviously, one type is called weaker than another, if it is implied by the latter. Hence, we can equip a collection of types of transitivity with this natural partial order relation and depict it graphically by means of a Hasse diagram.

A sufficient condition for verifying such an implication is the comparability of the corresponding upper bound functions: greater functions correspond to weaker types. However, a more refined investigation is often necessary to construct the full Hasse diagram. We have done this exercise for the types of transitivity discussed in this section.

The resulting Hasse diagram is shown in Figure 5.1. At the lower end of the diagram, $T_{\mathbf{M}}$-transitivity and multiplicative transitivity, two types of cycle-transitivity with respect to a self-dual upper bound function, are incomparable and can be considered as the strongest types of transitivity. At the upper end of the diagram, also $T_{\mathbf{L}}$-transitivity and weak stochastic transitivity are incomparable and can be considered as the weakest types of transitivity. Finally, note that the subchain consisting of partial stochastic transitivity, moderate product transitivity and weak product transitivity, bridges the gap between g-stochastic transitivity and T-transitivity.

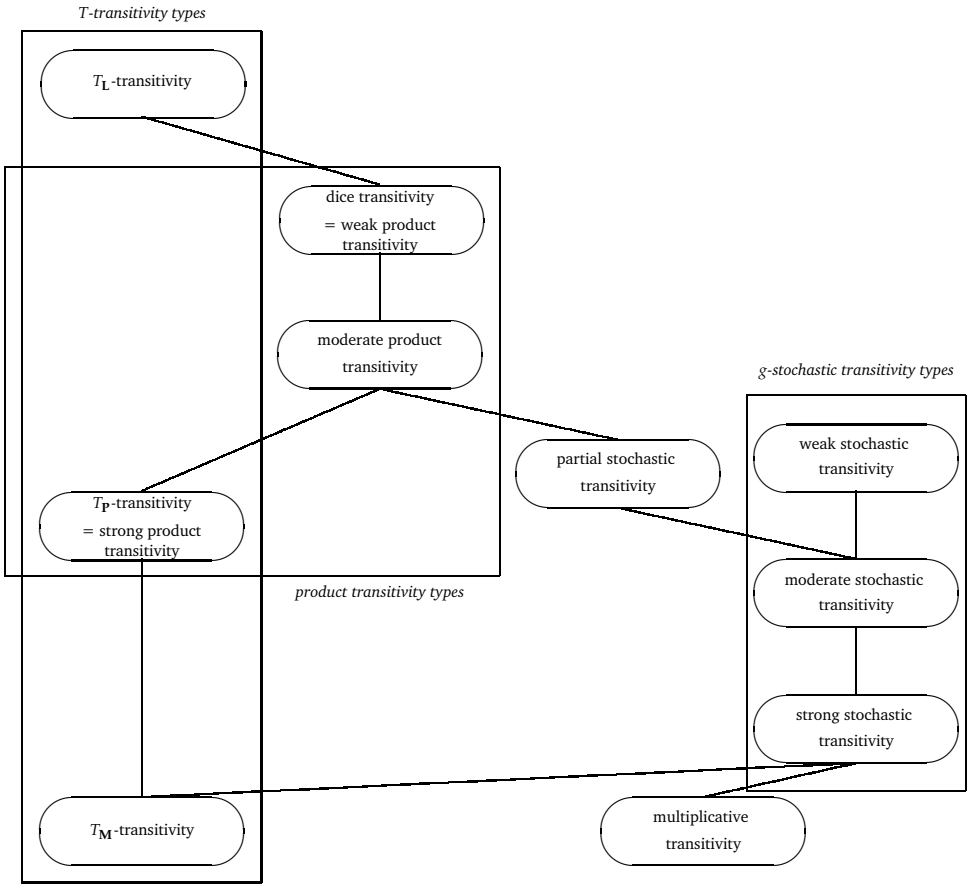


Figure 5.1: Hasse diagram with different types of cycle-transitivity.

5.4 Mutual rank probability relation

Given n random variables $X_1, X_2, \ldots, X_n$, we consider the reciprocal relation Q defined by

$$Q(X_i, X_j) = \text{Prob}(X_i > X_j) + \frac{1}{2}\,\text{Prob}(X_i = X_j).$$

If $X_1, X_2, \ldots, X_n$ are pairwisely independent, it has been proven in [46] that Q is dice transitive, irrespective of the marginal distributions of the random variables X_i. For discrete random variables, Q then describes the rules of a game with a collection of generalized dice (see also [47]). This explains the origin of the name dice transitivity.

If $X_1, X_2, \ldots, X_n$ are not pairwisely independent, the computation of Q requires the knowledge of the bivariate distribution functions. In our usual setting [35], these bivariate distribution functions are artificially constructed from the marginal distribution functions using a single copula C, inspired by the theorem of Sklar, *i.e.* $F_{X_i,X_j} = C(F_{X_i}, F_{X_j})$, for all random variables X_i, X_j. This artificial coupling need not be compatible with the random variables $X_1, X_2, \ldots, X_n$ constituting a random vector. For instance, as it is impossible for all (X_i, X_j) to be coupled counter-monotonically, the case $C = T_{\mathbf{L}}$ cannot occur in reality. This is related to the compatibility problem of copulas [92].

The following transitivity results have been obtained [35, 45, 48]:

(i) if $C = T_{\mathbf{M}}$, then the reciprocal relation Q is $T_{\mathbf{L}}$-transitive,

(ii) if $C = T_{\mathbf{L}}$, then the reciprocal relation Q is partially stochastic transitive.

Note that the first case might correspond to a realistic coupling of random variables, namely the situation in which the random variables are coupled comonotonically.

The set $\mathcal{E}(P)$ of all linear extensions of a finite poset $(P, \leq_P)$, equipped with the uniform measure $\mathcal{U}$, constitutes a probability space $(\mathcal{E}(P), \mathcal{P}(\mathcal{E}(P)), \mathcal{U})$. The discrete random variable $X_x : \mathcal{E}(P) \to \{1, \ldots, |P|\}$ then denotes the position (rank) $X_x(e)$ of an element $x \in P$ in a random linear extension

$e \in E_P$. Recall that the rank probability $\text{Prob}(\text{rank}(x) = i)$ is defined as the fraction of linear extensions of P in which x appears at position i, *i.e.*

$$\text{Prob}(\text{rank}(x) = i) = \text{Prob}(X_x = i).$$

The mutual rank probability $\text{Prob}(x > y)$ of two elements $x, y \in P$ is the fraction of linear extensions of P in which x succeeds y (x is ranked higher than y), *i.e.*

$$\text{Prob}(x > y) = \text{Prob}(X_x > X_y).$$

Let us define the *mutual rank probability relation* M_P, based on the mutual rank probabilities $\text{Prob}(x > y)$ of a poset $(P, \leq_P)$ as follows for all $x, y \in P$ (cf. Definition 4.3.1):

$$M_P(x, y) = \begin{cases} \text{Prob}(x > y) & \text{if } x \neq y, \\ 1/2 & \text{otherwise.} \end{cases} \tag{5.28}$$

Clearly, M_P is a reciprocal relation as it holds that

$$M_P(x, y) + M_P(y, x) = 1 \quad \forall x, y \in P. \tag{5.29}$$

We are concerned with the pairwise comparison of random variables constituting a discrete random vector $(X_{x_1}, \ldots, X_{x_n})$. Such a random vector is determined by a given finite poset $(P, \leq_P)$ where $P = \{x_1, \ldots, x_n\}$. The reciprocal relation M_P associated with such a random vector is nothing else but the mutual rank probability relation of the corresponding poset. Note that, despite the fact that the joint distribution function $F_{X_{x_1}, \ldots, X_{x_n}}$ does not lend itself to an explicit expression, a fair amount of pairwise couplings are of a very simple type. If it holds that $x <_P y$, then x precedes y in all linear extensions of P, whence X_x and X_y are comonotone. For pairs of incomparable elements, the bivariate couplings can vary from pair to pair. Certainly, these couplings cannot all be counter-monotone. Despite all this, it is possible to obtain transitivity results on mutual rank probability relations.

The problem of characterizing the transitivity of this mutual rank probability relation M_P, also called the problem of *proportional probabilistic transitivity* in a finite poset $(P, \leq_P)$, was already raised by Fishburn [60]: for any $u, v \in [0, 1]$, define $\delta(u, v)$ as

$$\delta(u, v) = \inf\{\text{Prob}(a > c) \mid \text{Prob}(a > b) \geq u \wedge \text{Prob}(b > c) \geq v\}, \tag{5.30}$$

where the infimum is taken over all finite posets $(P, \leq_P)$ and all $(a, b, c) \in P^3$. His problem can be elegantly reformulated as to identify the conjunctor $\delta : [0,1]^2 \to [0,1]$ such that the inequality

$$\delta(M_P(a,b), M_P(b,c)) \leq M_P(a,c) \tag{5.31}$$

holds for any finite poset $(P, \leq_P)$ and any $(a, b, c) \in P^3$. For obvious reasons, we will refer to δ-transitivity as *mutual rank transitivity*.

Fishburn proved that:

$$\begin{aligned}
&\delta(u,v) = 0 \quad \text{if } u+v<1\,,\\
&u+v-1 \leq \delta(u,v) \leq \min(u,v)\,,\\
&\delta(u,1-u) \leq 1/e\,,\\
&\delta(u,v) \leq 1-(1-u)(1-v)(1-\ln[(1-u)(1-v)])\,.
\end{aligned} \tag{5.32}$$

The conjunctor δ is clearly bounded from above by $T_{\mathbf{M}}$, expressing that mutual rank transitivity is weaker than $T_{\mathbf{M}}$-transitivity. On the other hand, δ is bounded from below by $T_{\mathbf{L}}$, implying that M_P is $T_{\mathbf{L}}$-transitive, a property that was shown to hold for any random vector in [35]. Hence, this is a very weak statement, and it is clear that M_P can be expected to exhibit a stronger type of transitivity. In other words, a sharper lower bound on δ is desired. Such a non-trivial lower bound on δ was obtained by Kahn and Yu [76] via geometric arguments.

Proposition 5.4.1. *For any $u, v \in [0,1]$, define $\delta^*(u,v)$ as*

$$\delta^*(u,v) = \inf\{\mathrm{Prob}(Y_i > Y_k) \mid \mathrm{Prob}(Y_i > Y_j) \geq u \wedge \mathrm{Prob}(Y_j > Y_k) \geq v\}\,,$$

where the infimum is taken over all $(Y_1, Y_2, \ldots Y_n)$ chosen uniformly from some n-dimensional compact convex subset of $\mathbb{R}^n$. Then it holds that

$$\delta^*(u,v) = \begin{cases} 0 & \text{if } u+v<1\,,\\ \min(u,v) & \text{if } u+v-1 \geq \min(u^2,v^2)\,,\\ \dfrac{(1-u)(1-v)}{u+v-2\sqrt{u+v-1}} & \text{otherwise}\,. \end{cases} \tag{5.33}$$

As Fishburn's problem can be embedded in the above more general setting, the following corollary was obtained [76].

Corollary 5.4.1. *The function δ^* provides a lower bound for δ, i.e. $\delta^* \leq \delta$.* $\diamond$

The above corollary implies that for any poset P and any $(a,b,c) \in P^3$, the mutual rank probability relation M_P satisfies the inequality

$$\delta^*(M_P(a,b), M_P(b,c)) \leq M_P(a,c)\,. \tag{5.34}$$

Since δ^* is a conjunctor, we can also say that mutual rank transitivity implies δ^*-transitivity. Obviously, δ^*-transitivity is stronger than $T_{\mathbf{L}}$-transitivity and weaker than $T_{\mathbf{M}}$-transitivity. Note that on the opposite diagonal

$$\{(u,v) \in [0,1]^2 \mid u+v=1\},$$

δ^* coincides with $T_{\mathbf{P}}$. However, in general the conjunctors $T_{\mathbf{P}}$ and δ^* are incomparable.

The conjunctor δ^* is also commutative, has neutral element 1 and is continuous on the set

$$\{(u,v) \in [0,1]^2 \mid u+v \geq 1\}.$$

However, it is not associative, and thus not a t-norm. Indeed,

$$\delta^*(0.5, \delta^*(0.6,0.7)) = 0.4150 \neq 0.4158 = \delta^*(\delta^*(0.5,0.6),0.7).$$

5.5 Situating δ^*-transitivity and mutual rank transitivity in the cycle-transitivity framework

We start this section with a positive result concerning δ^*-transitivity, namely that it implies moderate product transitivity. Since any mutual rank probability relation M_P is δ^*-transitive, Proposition 5.3.1 implies that M_P is cycle-transitive with respect to the upper bound function U_{δ^*}, given by

$$U_{\delta^*}(\alpha,\beta,\gamma) = \min(V(\alpha,\beta), V(\alpha,\gamma), V(\beta,\gamma))\,, \tag{5.35}$$

where the function $V : \Delta_2 \to [0,1]$ is defined by

$$V(u,v) = \begin{cases} u+v & \text{if } u+v<1\,, \\ v & \text{if } u+v-1 \geq u^2\,, \\ u+v-\dfrac{(1-u)(1-v)}{(1-\sqrt{u+v-1})^2} & \text{if } u \leq v \leq 1-u+u^2 \leq 1\,. \end{cases} \tag{5.36}$$

Note that each of $V(\alpha,\beta)$, $V(\alpha,\gamma)$ and $V(\beta,\gamma)$ can be considered as an upper bound function.

Theorem 5.5.1. *δ^*-transitivity implies moderate product transitivity.*

Proof : One way of proving this theorem would be to show that $U_{\delta^*}(\alpha,\beta,\gamma) \leq U_{mp}(\alpha,\beta,\gamma)$. Unfortunately, there is no fixed ordering on the function values $V(\alpha,\beta)$, $V(\alpha,\gamma)$ and $V(\beta,\gamma)$ for all $(\alpha,\beta,\gamma) \in \Delta_3$. In order to circumvent this problem, we will prove that cycle-transitivity with respect to each of the upper bound functions $V(\alpha,\beta)$, $V(\alpha,\gamma)$ and $V(\beta,\gamma)$ implies moderate product transitivity. Hence, the proof consists of three parts. For technical reasons we prefer to use the equivalent upper bound function U'_{mp} for moderate product transitivity, given by

$$U'_{mp}(\alpha,\beta,\gamma) = \begin{cases} \alpha+\gamma-\alpha\gamma & \text{if } \beta \geq 1/2\,, \\ 2 & \text{if } \beta < 1/2\,. \end{cases} \tag{5.37}$$

(i) *Cycle-transitivity with respect to $V(\beta,\gamma)$ implies moderate product transitivity*

We need to prove for any $(\alpha,\beta,\gamma) \in \Delta_3$ that $\alpha+\beta+\gamma-1 \leq V(\beta,\gamma)$ implies that $\alpha+\beta+\gamma-1 \leq U'_{mp}(\alpha,\beta,\gamma)$.

If $\beta+\gamma < 1$ (and hence $\beta < 1/2$), the implication is trivially fulfilled. This is also the case when $\beta+\gamma-1 \geq \beta^2$ since then $\gamma \leq U'_{mp}(\alpha,\beta,\gamma)$. Next, we investigate the remaining case $\beta \leq \gamma \leq 1-\beta+\beta^2 \leq 1$. Then, $\alpha+\beta+\gamma-1 \leq V(\beta,\gamma)$ is equivalent to

$$\alpha \leq 1 - \frac{(1-\beta)(1-\gamma)}{(1-\sqrt{\beta+\gamma-1})^2}\,,$$

whereas $\alpha+\beta+\gamma-1 \leq U'_{mp}(\alpha,\beta,\gamma)$ is equivalent to

$$\alpha \leq \frac{1-\beta}{\gamma}\,.$$

Obviously, we must prove that

$$1 - \frac{(1-\beta)(1-\gamma)}{(1-\sqrt{\beta+\gamma-1})^2} \leq \frac{1-\beta}{\gamma}\,, \tag{5.38}$$

or, equivalently, that the inequality

$$\beta\gamma^2 - (\beta+\gamma-1)(\beta+2\gamma-2\sqrt{\beta+\gamma-1}) \geq 0\,,$$

holds for all $\gamma \leq 1-\beta+\beta^2$. To verify this, let us regard the left-hand side of this inequality as a function of two independent variables β and γ and compute its extremal points. These are solutions of the system

$$\begin{cases} \beta+\gamma-1+\gamma^2-2\beta\gamma = 0 \\ 3\beta+4\gamma-2\beta\gamma-2-3\sqrt{\beta+\gamma-1} = 0\,. \end{cases}$$

There is only one real solution $(\beta,\gamma)=(1,1)$, but computation of the Hessian proves that it is not a minimizer. Therefore, the minimum of this function on the domain under consideration must be attained on its border. This border consists of 3 segments:

(a) The curve $\gamma = 1-\beta+\beta^2$ with $\beta \in [1/2,1]$. The function value on the curve is $\beta(\gamma-\beta)^2$, and is always positive.

(b) The line segment $\beta = 1/2$ with $\gamma \in [1,2,3/4]$. The function value on the line segment is $2(\gamma-1/2)\sqrt{\gamma-1/2} - 3\gamma^2/2 + \gamma/2 + 1/4$, and it is easily verified that on the interval $[1/2,3/4]$ this function attains its minimum value $1/32$ at $\gamma = 3/4$.

(c) The line segment $\gamma = \beta$ with $\beta \in [1/2,1]$. The function reduces to $\beta^3 - (2\beta-1)(3\beta-2\sqrt{2\beta-1})$, and is always positive.

(ii) *Cycle-transitivity with respect to $V(\alpha,\gamma)$ implies moderate product transitivity*

Now both $V(\alpha,\gamma)$ and $U'_{mp}(\alpha,\beta,\gamma)$ are functions of α and γ alone. If $\alpha+\gamma<1$, then cycle-transitivity with respect to $V(\alpha,\gamma)$ imposes no condition $\beta < 1/2$ and it also follows that $\beta \leq \gamma < 1-\alpha$, whence $\alpha+\beta+\gamma-1 < \alpha+\gamma-\alpha\gamma$, showing that cycle-transitivity with respect to $U'_{mp}(\alpha,\beta,\gamma)$ imposes no restriction. If $\alpha+\gamma-1 \geq \alpha^2$, the inequality $V(\alpha,\gamma) \leq U'_{mp}(\alpha,\beta,\gamma)$ holds as $\gamma \leq U'_{mp}(\alpha,\beta,\gamma)$ everywhere in Δ_3. There remains to investigate the domain determined by $\gamma \leq 1-\alpha+\alpha^2 \leq 1$. We need to prove that on this domain the inequality

$$\frac{(1-\alpha)(1-\gamma)}{(1-\sqrt{\alpha+\gamma-1})^2} - \alpha\gamma \geq 0\,,$$

or, equivalently, the inequality

$$4\alpha^2\gamma^2 - (\alpha+\gamma-1)(\alpha\gamma+1)^2 \geq 0\,,$$

holds. The left-hand side of the latter, regarded as a function of two independent variables α and γ, has only one extremal point in $[0,1]^2$, namely $(\alpha,\gamma) = (1,1)$. It is a point of the domain under consideration and the function attains there the value 0. However, this point does not yield a minimizer and we have to study the function on the border of the domain. The reader can easily verify that on that border the minimum is indeed attained at the point $(\alpha,\gamma) = (1,1)$.

(iii) *Cycle-transitivity with respect to $V(\alpha,\beta)$ implies moderate product transitivity*

The proof goes as for the other cases. The only non-trivial situation occurs in the subdomain determined by $\alpha \leq \beta \leq 1-\alpha+\alpha^2$. On this subdomain cycle-transitivity with respect to $V(\alpha,\beta)$ is equivalent to the inequality

$$\gamma \leq 1 - \frac{(1-\alpha)(1-\beta)}{(1-\sqrt{\alpha+\beta-1})^2}\,,$$

whereas cycle-transitivity with respect to U'_{mp} is equivalent to the inequality

$$\gamma \leq \frac{1-\beta}{\alpha}\,.$$

Hence, we must prove that the inequality

$$1 - \frac{(1-\alpha)(1-\beta)}{(1-\sqrt{\alpha+\beta-1})^2} \leq \frac{1-\beta}{\alpha}\,,$$

or, equivalently, the inequality

$$\alpha(1-\alpha)(1-\beta) - (\alpha+\beta-1)(\alpha+\beta-2\sqrt{\alpha+\beta-1}) \geq 0\,,$$

holds for all $\alpha \leq \beta \leq 1-\alpha+\alpha^2$. It can be shown that the left-hand side, regarded as a function of α and β, has no minimizer inside the domain under consideration. The minimum is attained on the border of the domain at all points of the curve $\beta = 1-\alpha+\alpha^2$ where $\alpha \in [0,1/2]$. Since the minimum function value is 0, the inequality indeed holds on that domain. ∎

Theorem 5.5.1 states that δ^*-transitivity and mutual rank transitivity have to be situated below moderate product transitivity in Figure 5.3. This allows us to conclude that:

(i) Weak stochastic transitivity does neither imply δ^*-transitivity, nor mutual rank transitivity.

We continue with some negative results.

Proposition 5.5.1. *Moderate stochastic transitivity does not imply δ^*-transitivity.*

Proof : Consider the reciprocal relation Q on $A = \{a, b, c\}$ defined by $Q(a, b) = 0.4$, $Q(b, c) = 0.6$ and $Q(a, c) = 0.2$. Obviously, Q is moderately stochastic transitive, since $\min(Q(b, c), Q(c, a)) = 0.6 \geq Q(b, a) = 0.6$. However, since $\delta^*(Q(a, b), Q(b, c)) = 0.24 > Q(a, c) = 0.2$, Q is not δ^*-transitive. ∎

Proposition 5.5.1 generates the following additional results:

(ii) Moderate stochastic transitivity does not imply mutual rank transitivity.

(iii) Partial stochastic transitivity does neither imply δ^*-transitivity, nor mutual rank transitivity.

Proposition 5.5.2. *Mutual rank transitivity does not imply partial stochastic transitivity, nor weak stochastic transitivity.*

Proof : Partial stochastic transitivity of a reciprocal relation states that $\alpha + \beta \leq 1$ for all $\beta \in [0, 1]$, which implies that $\alpha \leq 1/2$. In particular, if a reciprocal relation Q on A is partially stochastic transitive, no triplet $(a, b, c) \in A^3$ can exist, such that simultaneously $Q(a, b) > 1/2$, $Q(b, c) > 1/2$, and $Q(c, a) > 1/2$. The same situation is incompatible with weak stochastic transitivity.

Now consider the poset with Hasse diagram in Figure 5.2. For the elements a, b and c it holds that $M_P(b, a) = M_P(a, c) = M_P(c, b) = 80/159 > 1/2$,

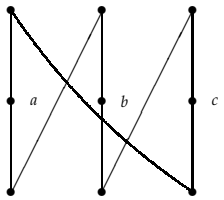


Figure 5.2: A poset where $M_P(b,a) = M_P(a,c) = M_P(c,b) > 1/2$.

which is a so-called linear extension majority cycle. Note that such linear extension cycles will be the subject of Chapter 6. Since the above situation is encountered, it leads to the announced negative results. ∎

Proposition 5.5.2 generates the following additional results:

(iv) δ^*-transitivity does not imply partial stochastic transitivity, nor weak stochastic transitivity.

(v) Neither mutual rank transitivity, nor δ^*-transitivity, implies any type of transitivity stronger than partial or weak stochastic transitivity, such as moderate and strong stochastic transitivity.

Next, we provide a second positive result.

Proposition 5.5.3. *Strong stochastic transitivity implies mutual rank transitivity.*

Proof : In [38] the t-norm T_{ss} defined by

$$T_{ss}(x,y) = \begin{cases} \min(x,y) & \text{if } \max(x,y) \geq 1/2\,, \\ 0 & \text{if } \max(x,y) < 1/2\,, \end{cases}$$

is introduced, which is situated between $T_{\mathbf{L}}$ and $T_{\mathbf{M}}$. It is shown that the upper bound function $U_{T_{ss}}$ obtained from (5.25) by setting $f = T_{ss}$ is equivalent to the upper bound function that characterizes strong stochastic transitivity. Hence, T_{ss}-transitivity is equivalent to strong stochastic transitivity.

Since $\delta(x,y) = 0$ if $x + y < 1$ and $\delta \leq \min$ it holds that $\delta \leq T_{ss}$, whence strong stochastic transitivity implies mutual rank transitivity. ∎

Finally, we need to clarify the relationship between $T_{\mathbf{P}}$-transitivity on the one hand, and δ^*-transitivity and mutual rank transitivity on the other hand.

Proposition 5.5.4. *Mutual rank transitivity does not imply $T_{\mathbf{P}}$-transitivity.*

Proof : Consider the poset $P = \{a,b,c\}$, with $<_P = \{(a,c)\}$. One easily verifies that $M_P(a,b) = 1/3$, $M_P(b,c) = 1/3$ and $M_P(a,c) = 0$, which violates $T_{\mathbf{P}}$-transitivity. ∎

Proposition 5.5.5. *$T_{\mathbf{P}}$-transitivity does not imply δ^*-transitivity.*

Proof : Consider the reciprocal relation Q on $A = \{a,b,c\}$ defined by $Q(a,b) = 0.6$, $Q(b,c) = 0.6$ and $Q(a,c) = 0.4$. Obviously, Q is $T_{\mathbf{P}}$-transitive. However, since

$$\delta^*(Q(a,b),Q(b,c)) = 0.5236 > Q(a,c) = 0.4,$$

Q is not δ^*-transitive. ∎

Propositions 5.5.4 and 5.5.5 generate the following additional results:

(vi) δ^*-transitivity does not imply $T_{\mathbf{P}}$-transitivity.

(vii) $T_{\mathbf{P}}$-transitivity does not imply mutual rank transitivity.

Combining all of the above results leads to the precise location of δ^*-transitivity and mutual rank transitivity in the Hasse diagram of Figure 5.1, leading to the extended Hasse diagram in Figure 5.3.

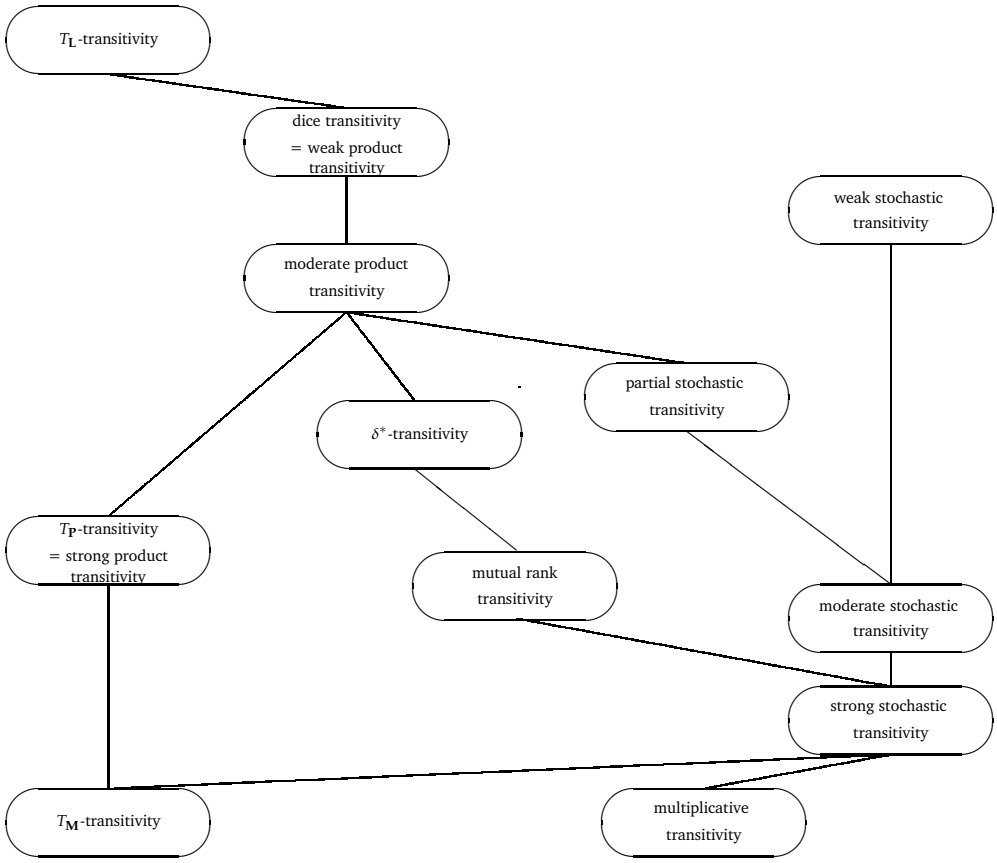


Figure 5.3: Hasse diagram including δ^*-transitivity and mutual rank transitivity.

6 Linear extension majority cycles

6.1 Counting posets with linear extension majority cycles

Definition 6.1.1. The *linear extension majority relation* or *LEM relation* of a poset P is the binary relation $\succ_{\text{LEM}}$ on P such that $x \succ_{\text{LEM}} y$ if $\text{Prob}(x > y) > \text{Prob}(y > x)$. Since the mutual rank probability relation is reciprocal, it is equivalent to say that $x \succ_{\text{LEM}} y$ if $\text{Prob}(x > y) > 1/2$. $\diamond$

The linear extension majority relation $\succ_{\text{LEM}}$ first appeared in 1968 in the work of Kislitsyn [78], and it was conjectured that $\succ_{\text{LEM}}$ is transitive, and thus cannot contain cycles, *i.e.* subsets $\{x_1, x_2, \ldots, x_m\}$ of elements of P such that $x_1 \succ_{\text{LEM}} x_2 \succ_{\text{LEM}} \cdots \succ_{\text{LEM}} x_m \succ_{\text{LEM}} x_1$. However, in 1974 Fishburn [58] showed that $\succ_{\text{LEM}}$ can contain cycles, and thus is not transitive. These cycles are referred to as *LEM cycles* of length m. Since then, quite some attention has been given to LEM cycles in the literature. Examples of posets with LEM cycles in different contexts are given in [1, 59, 60, 61, 62, 67, 69], frequency estimates for LEM cycles have been reported in [66, 68], and the occurrence of LEM cycles in certain subclasses of posets has been studied in [14, 54].

Aside from the fact that the existence of LEM cycles is an intriguing phe-

nomenon in its own right, a better understanding of LEM cycles might help in the ongoing quest to characterize the transitivity of mutual rank probabilities in posets [37, 60, 76, 121]. Furthermore, Gehrlein and Fishburn [68] discuss an interesting application of LEM cycles in which incomplete information about a linear order $\leq_L$ on a set P is given in the form of a partial order $\leq_P$. Assuming the partial information is correct, they consider the problem of attempting to reconstruct the linear order $\leq_L$. The choice of such a linear order amounts to the selection of a single extension from the set of all linear extensions of the poset. As already described in Chapter 4, this is a problem frequently encountered in real world situations, e.g. when a decision maker insists on obtaining a linear order on all objects instead of a partial order obtained by comparing the attribute vectors of the objects [22, 24, 28, 40, 44, 83, 84, 96]. The approach Gehrlein and Fishburn suggest assesses a conditional probability p_i that the corresponding linear extension $(P, \leq_{L_i})$ of the poset represents $\leq_L$ given the partial information contained in $\leq_P$. Once these probabilities p_i are obtained, it is possible to compute the probability that $x <_L y$ as the sum of all probabilities p_i corresponding to a linear extension for which it holds that $x <_{L_i} y$.

One model of interest describing the manner in which the partial order $\leq_P$ is obtained from $\leq_L$ implies that all of these probabilities p_i equal $1/e(P)$. In this case, the probability that $x <_L y$ is identical to the mutual rank probability $\text{Prob}(y > x)$ according to the partial order $\leq_P$. Moreover, the maximum likelihood estimator $\prec^*$ of $\leq_L$ defined by $x \prec^* y$ if $\text{Prob}(y > x) > \text{Prob}(x > y)$, is nothing else but the LEM relation, in the sense that $x \prec^* y$ if and only if $y \succ_{\text{LEM}} x$. Since posets exist with LEM cycles, the maximum likelihood estimator $\prec^*$ can be intransitive. This notion of a maximum likelihood estimator of $\leq_L$ being quite appealing, it would be interesting to obtain some measure of the propensity of this technique to produce intransitive maximum likelihood estimators.

Gehrlein and Fishburn [69] conducted a computer search to find all non-isomorphic posets with LEM cycles for poset cardinalities $n \leq 9$ and showed that no cycles exist for $n \leq 8$. Moreover, exactly 5 non-isomorphic posets for $n = 9$ were found. In Figure 6.1 three of these posets are depicted. The other posets are their dual versions. Remark however that due to symmetry the leftmost poset coincides with its dual, such that 5 unique non-isomorphic posets with LEM-cycles are obtained for $n = 9$. In a later paper, Gehrlein [66] estimated the likelihood of LEM cycles for posets on up to $n = 12$ elements by generating random connected posets.

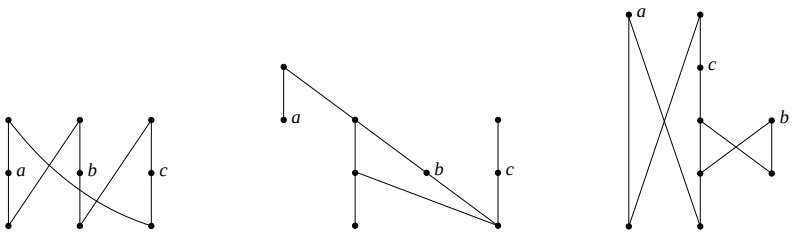


Figure 6.1: Three posets of size 9 with LEM-cycles ($a \succ_{LEM} b$, $b \succ_{LEM} c$ and $c \succ_{LEM} a$).

6.1.1 The algorithm

Unless some direct method is invented to avoid explicit enumeration, counting all posets with LEM cycles requires at least all posets to be enumerated and their mutual rank probability relation to be computed.

Brinkmann and McKay [18] developed a very efficient method to construct pairwise non-isomorphic posets, which allows them to enumerate posets on up to 16 elements. As an illustration of the size of the problem, the number of non-isomorphic posets of sizes 9 to 16 are shown in Table 6.1.

n	number of posets
9	183 231
10	2 567 284
11	46 749 427
12	1 104 891 746
13	33 823 827 452
14	1 338 193 159 771
15	68 275 077 901 156
16	4 483 130 665 195 087

Table 6.1: Number of non-isomorphic posets for $n = 9, 10, \ldots, 16$.

As shown in Subsection 4.3.1, a direct way to compute the mutual rank probability relation using the lattice of ideals representation of a poset no longer necessitates the enumeration of all linear extensions, though requires additional memory for storing the lattice of ideals. Since we are precisely interested in generating small posets, the lattice of ideals of such posets

nicely fits into memory of current computer architectures.

Indeed, an approach based on enumerating all linear extensions would imply that only for the antichain of 13 elements, already more than 62 billion linear extensions need to be enumerated. This computationally extremely intensive process combined with the number of posets that grows quickly for increasing n, would make the counting procedure out of reach for $n = 12$, let alone for $n = 13$.

We combined the poset generation algorithm of Brinkmann and McKay [18] and Algorithm 4.2 described in Subsection 4.3.1 to compute the mutual rank probability relation for each poset enumerated. This approach enabled us to obtain exact counts for posets on up to 13 elements.

For each poset $(P, \leq_P)$ generated by the algorithm of Brinkmann and McKay, Algorithm 6.1 is executed. This algorithm checks whether a given poset $(P, \leq_P)$ contains a LEM cycle of length l, where $l = 3, 4, \ldots, k$ with $k \leq n$. The result of the check is returned as an array of booleans in which the element at index l is true if $(P, \leq_P)$ contains a LEM cycle of length l and false in the negative case.

In the first line of Algorithm 6.1, the Hasse diagram L_P of the lattice of ideals of $(P, \leq_P)$ is constructed. The most efficient algorithm currently known for constructing is the algorithm of Habib *et al.* [70] presented in Section 2.3, which has an optimal complexity up to a constant factor. In line number 2 the mutual rank probability relation M_P is computed as in Algorithm 4.2.

Let us construct the directed weighted graph G in which the vertices are the elements of our poset $(P, \leq_P)$ and the directed edges the couples $(x, y) \in P^2$ for which $M_P(x, y) > 1/2$. Furthermore, let us attribute a weight $M_P(x, y)$ to each directed edge (x, y). Clearly, a cycle in G of length l is a LEM cycle of length l.

Once the relation M_P is computed, in order to know whether there exists a closed walk in G from a vertex $x \in V$ to the same vertex x with l directed edges, we calculate M_P^l, in which the matrix multiplication $M_P^{l-1} \times M_P$ is defined as the usual matrix multiplication where min is substituted for $\cdot$ and max for $+$. Remark that the first multiplication $M_P \times M_P$ assigned to M_P^2 takes place before the for-loop in line 3, since it is impossible for cycles of length 2 to occur due to the reciprocity of the mutual rank probability

Algorithm 6.1 Checking whether a given poset $(P, \leq_P)$ contains a LEM cycle of length l, where $l = 3, 4, \ldots, k$ with $k \leq n$

build the Hasse diagram L_P of the lattice of ideals $(\mathcal{I}(P), \subseteq)$ of $(P, \leq_P)$
compute the mutual rank probability relation M_P using $(\mathcal{I}(P), \subseteq)$
$M_P^2 \leftarrow M_P \times M_P$
for each $l = 3, 4, \ldots, k$ **do**
    cycle$[l] \leftarrow$ **false**
    $M_P^l \leftarrow M_P^{l-1} \times M_P$
    $c \leftarrow 0$
    for each $j = 1, \ldots, n$ **do**
        if $M_P^l(j, j) > 1/2$ **then**
            if $l < 6$ **then**
                cycle$[l] \leftarrow$ **true**
                break for
            $c \leftarrow c + 1$
            elem$[c] \leftarrow j$
    if $c \geq l$ **then**
        for each $i = 1, 2, \ldots, c$ **do**
            if there is a cycle of length l starting in elem$[i]$ **then**
                cycle$[l] \leftarrow$ **true**
                break for
return cycle

relation M_P. Each subsequent multiplication is executed inside the for-loop in line 6. If it holds for an element x that $M_P^l(x,x) > 1/2$ (line 9), there is a closed walk with l edges. It is clear that, if there is no closed walk with l edges, a cycle of length l is impossible, so no further check is needed. If there is a closed walk of length $l < 6$, there is a cycle of length l (line 10). Indeed, due to the fact that if an edge $(x,y) \in E$ is present, no edge (y,x) can be present, the smallest cycles that can occur have length 3. For $l = 6$, a closed walk of length 6 could arise from two cycles of length 3 sharing one common vertex. Therefore it is clear that for $l \geq 6$ a situation can occur in which a closed walk of length l is the composition of two or more cycles.

Since we are searching for cycles of length l, it is clear that, in order for such a cycle to occur, at least l elements should have a closed walk of length l (line 15). Finally, an explicit depth-first search should be done for each candidate element (line 17), *i.e.* each element x having $M_P^l(x,x) > 1/2$, to verify whether a cycle of length l is present. For each such candidate, recursively all possible successor edges which have not yet been visited are selected until exactly l edges have been chosen. Subsequently, if the vertex at the last edge is the starting vertex, a cycle is detected.

The algorithm we implemented in Java is a actually slight variant of the above one, in the sense that we search for all possible LEM cycles of length l instead of just returning whether $(P, \leq_P)$ contains a LEM cycle of length l. Moreover, each poset in which a LEM cycle occurs is stored in a database for future reference.

6.1.2 Results

When generating posets on n elements, the algorithm of Brinkmann and McKay [18] will, for arbitrary $r, m \in \mathbb{N}, r < m$, generate all posets on $n-4$ elements and number them in the order they occur, while only generating successors of those posets whose number equals $r \mod m$. This option allowed us to split the generation process, and thus the counting procedure. For $n = 12$, we divided the generation process into 100 parts and for $n = 13$ into 1000 parts. Rescaled to a single 2.4 GHz processor the entire process for $n = 13$ would take around 4 computing years. Because of the fact that the number of posets for $n = 14$ is almost 40 times larger than the number of posets for $n = 13$, combined with the exponential behaviour in n of the

number of ideals, it is not possible to obtain results for $n > 13$ in a feasible time frame with our approach, unless substantially more computing power is available. Moreover, due to the growing size of the lattice of ideals, the size of the memory also becomes a constraining factor for larger values of n. It should be remarked that implementing Algorithm 6.1 in a language which is more performant than Java could possibly reduce the running time in a substantial way. However, due to the exponential nature of the problem it would now allow us to go any step further in the computation: counting all LEM cycles on posets on up to 14 elements would still be out of reach.

In order to verify the correctness of the implementation of Algorithm 4.2, for posets on up to 9 elements all mutual rank probability relations were compared with the results obtained by an independent approach based on the Varol-Rotem algorithm listed in Algorithm 3.1. For each poset, all linear extensions were enumerated and the mutual rank probability relation was computed.

6.1.2.1 LEM cycles

$n\backslash l$	3	4	5	6	7	8	all
9	5	-	-	-	-	-	5
10	148	6	-	-	-	-	153
11	5 740	101	-	-	-	-	5 815
12	216 573	2 885	5	21	-	-	218 097
13	9 318 881	102 127	471	363	1	-	9 348 400

Table 6.2: Number of n-element posets with LEM cycles of length l, for $n = 9, 10, \ldots, 13$ and $l = 3, 4, \ldots, 8$.

In Table 6.2, the number of n-element posets having a LEM cycle of length l is shown, while in Table 6.3 the relative number of n-element posets having a LEM cycle of length l, multiplied by 10^4, is shown. The fact that for $n > 9$, the numbers in the last column in each of these tables are smaller than the sum of the preceding numbers on the same row, reflects the fact that some posets contain LEM cycles of different lengths.

Clearly, the results in Table 6.3 provide additional support for the conjecture formulated by Gehrlein and Fishburn [69] that the likelihood of observing

$n\backslash l$	3	4	5	6	7	8	all
9	0.273	-	-	-	-	-	0.273
10	0.576	0.023	-	-	-	-	0.596
11	1.228	0.022	-	-	-	-	1.244
12	1.960	0.026	0.000	0.000	-	-	1.974
13	2.755	0.030	0.000	0.000	0.000	-	2.764

Table 6.3: Relative number of n-element posets having a LEM cycle of length l, multiplied by 10^4, for $n = 9, 10, \ldots, 13$ and $l = 3, 4, \ldots, 8$.

a random poset with a LEM cycle increases as n increases. The rate at which the likelihood increases, however, seems to decrease with increasing n. This is in accordance with the LEM cycle frequency estimates made by Gehrlein [66]. Moreover, when we sampled 1 071 200 posets of size 14 uniformly at random, a relative number of $3.641 \cdot 10^{-4}$ turned out to have at least one cycle, in line with the trend of a decreasing rate.

When using nonlinear optimization to fit a function $f(x) = a + b \cdot x + x \cdot \log(x)$ with two parameters $a, b \in \mathbb{R}$ to the total relative frequencies in Table 6.3, we found values $a = 5.2304$ and $b = -2.7572$ such that $f(x)$ explains 99.59% of the variance. Of course, using such a formula to estimate the relative number of n-element posets with LEM cycles should be done with great caution, especially for larger n. We expect, however, that for n smaller than 50, a reasonable approximation is obtained using $f(x)$.

6.1.2.2 Posets of height 1

Ewacha et al. [54] have shown that posets of height 1 can have LEM cycles. Our results indicate that the smallest posets having this property have 11 elements. Actually, there are only two such 11-element posets: the poset depicted in Figure 6.2 and its dual poset which have a LEM cycle of length 4. The results of this counting operation for $n = 11, 12, 13$ are shown in Table 6.4, while the relative number of n-element posets of height 1 having LEM cycles are shown in Table 6.5. Remark that no 11-element poset of height 1 has LEM cycles of length 3. An analogous observation can be made for 12 and 13-element posets of height 1: although LEM cycles of length 6 occur, no poset has a LEM cycle of length 5. It is clear that the probability

of encountering a LEM cycle in the class of n-element posets of height 1 is much lower than in the class of all n-element posets. Note also that the relative number of posets of height 1 that have a LEM cycle seems to increase at a lower pace than the total number of n-element posets that have a LEM cycle.

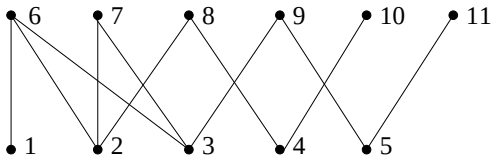


Figure 6.2: The smallest poset of height 1 having a LEM cycle, where $\text{Prob}(4 > 2) = \text{Prob}(5 > 3) = 174660/349260$ and $\text{Prob}(3 > 4) = \text{Prob}(2 > 5) = 174790/349260$

$n\backslash l$	3	4	5	6	7	all
11	-	2	-	-	-	2
12	11	9	-	1	-	20
13	175	123	-	3	-	296

Table 6.4: Number of n-element posets of height 1 having a LEM cycle of length l, for $n = 11, 12, 13$ and $l = 3, 4, \ldots, 7$.

$n\backslash l$	3	4	5	6	7	all
11	-	0.308	-	-	-	0.308
12	0.219	0.179	-	0.020	-	0.399
13	0.345	0.273	-	0.006	-	0.584

Table 6.5: Relative number of n-element posets of height 1 having a LEM cycle of length l, multiplied by 10^4, for $n = 11, 12, 13$ and $l = 3, 4, \ldots, 7$.

6.1.2.3 Worst balanced posets

For a poset $(P, \leq_P)$, the *balance constant* $b(P)$ is defined as the maximum over all pairs $(x, y) \in P$ of $\min(\text{Prob}(x > y), \text{Prob}(y > x))$. *Worst balanced* n-element posets are n-element posets of which the balance constant is the

smallest and which are not a linear sum of other posets. The importance of these worst balanced posets stems from a well-known conjecture made by Kislitsyn in 1968 [78], known as the $1/3$-$2/3$-*conjecture*. It states that in any non-chain poset P one can always find a couple of elements $(x, y) \in P$ such that

$$1/3 \leq \text{Prob}(x > y) \leq 2/3.$$

Brightwell et al. [13] proved that there exists a couple of elements $(x, y) \in P$ such that

$$(5 - \sqrt{5})/10 \leq \text{Prob}(x > y) \leq (5 + \sqrt{5})/10$$

and showed that for a class of countably infinite posets for which the notion of mutual rank probabilities makes sense, it is the best possible bound. A finite non-chain poset for which $b(P) < 1/3$ would be a counterexample to this conjecture.

Brightwell [12] presented all worst balanced posets for n up to 8, and Peczarski [98] found the worst balanced posets for $n = 9, 10, 11$. Due to the regularity one can observe, Peczarski introduced a new class of badly balanced posets, which he called *ladders with broken rungs*. We obtained the worst balanced posets for $n = 12, 13$, and, as can be seen in Figure 6.3, they indeed fall into Peczarski's class of ladders with broken rungs.

6.2 Minimum cutting levels

In the previous section it was shown that the mutual rank probability relation can contain LEM cycles, and all posets on up to 13 elements containing such cycles were counted. We would now like to determine a minimum cutting level δ_m such that the crisp relation obtained from the mutual rank probability relation by setting its elements smaller than or equal to δ_m equal to 0 and its other elements equal to 1 is free from cycles of size $l \leq m$. In other words, we want to obtain the minimum δ_m such that at least one mutual rank probability in any LEM cycle of size $l \leq m$ is smaller than or equal to δ_m.

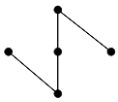
Definition 6.2.1. The *strict cut* at $\delta \in [1/2, 1[$ of a reciprocal relation Q

P_3

$b(P_3) = 0.333333\ldots$

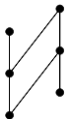
P_4

$b(P_4) = 0.4$



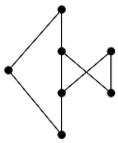
P_5

$b(P_5) = 0.363636\ldots$



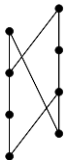
P_6

$b(P_6) = 0.357142\ldots$



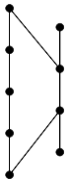
P_7

$b(P_7) = 0.358974\ldots$



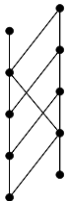
P_8

$b(P_8) = 0.355555\ldots$



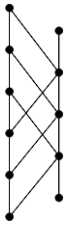
P_9

$b(P_9) = 0.352941\ldots$



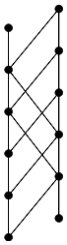
P_{10}

$b(P_{10}) = 0.349057\ldots$



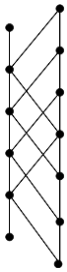
P_{11}

$b(P_{11}) = 0.350877\ldots$



P_{12}

$b(P_{12}) = 0.350181\ldots$



P_{13}

$b(P_{13}) = 0.350446\ldots$

Figure 6.3: All worst balanced posets for $n = 3, 4, \ldots, 13$.

defined on a set A, is the crisp relation Q^δ defined by

$$Q^\delta(x,y) = \begin{cases} 1 & \text{if } Q(x,y) > \delta\,, \\ 0 & \text{otherwise.} \end{cases}$$

◇

Definition 6.2.2. The *minimum cutting level* δ_m is the smallest number such that for any finite poset the strict cut at δ_m of the corresponding mutual rank probability relation is free of LEM cycles of length $l \leq m$. ◇

Recall that in the previous section a directed weighted graph G was introduced in which the vertices are the elements of our poset $(P, \leq_P)$ and the directed edges the couples $(x,y) \in P^2$ for which $\text{Prob}(x > y) > 1/2$. Finding a minimal cutting level δ_m is equivalent to requiring the graph obtained by removing edges from G corresponding to values $\text{Prob}(x > y) \leq \delta_m$ to be free of cycles of length $l \leq m$.

6.2.1 Theoretical upper bounds

From the previous chapter it is known that the mutual rank probability relation M_P of any poset $(P, \leq_P)$ is cycle-transitive with respect to the upper bound function

$$U(\alpha,\beta,\gamma) = \alpha + \gamma - \alpha\gamma. \tag{6.1}$$

Using the definition of cycle-transitivity (5.22) this is identical to imposing that for any $a,b,c \in P$ it holds that

$$\alpha + \beta + \gamma - 1 \leq \alpha + \gamma - \alpha\gamma, \tag{6.2}$$

or equivalently

$$1 - \beta \geq \alpha\gamma, \tag{6.3}$$

where α, β and γ are defined as usually, *i.e.*

$$\begin{aligned} \alpha &= \min\left(M_P(a,b), M_P(b,c), M_P(c,a)\right), \\ \beta &= \text{median}\left(M_P(a,b), M_P(b,c), M_P(c,a)\right), \\ \gamma &= \max\left(M_P(a,b), M_P(b,c), M_P(c,a)\right). \end{aligned}$$

This property allows us to theoretically derive an upper bound for the minimum cutting level δ_m.

Denote by $\mathcal{MR}$ the set of mutual rank probability relations M_P of all finite posets $(P, \leq_P)$ and by $\mathcal{CR}$ the set of all reciprocal relations Q on a finite set A that are cycle-transitive with upper bound function (6.1). Clearly $\mathcal{MR} \subseteq \mathcal{CR}$.

First the case $m = 3$ is considered. The minimum cutting level δ_3 is the largest value that can be obtained for the minimum of $M_P(a,b)$, $M_P(b,c)$ and $M_P(c,a)$ forming a cycle of length 3 in a mutual rank probability relation M_P of a poset $(P, \leq_P)$ with $a, b, c \in P$, *i.e.*:

$$\delta_3 = \sup_P \max_{a,b,c \in P} \min(M_P(a,b), M_P(b,c), M_P(c,a)),$$

where the supremum is taken over all finite posets $(P, \leq_P)$. Since $\mathcal{MR} \subseteq \mathcal{CR}$, an upper bound $\bar{\delta}_3$ for δ_3 is

$$\bar{\delta}_3 = \sup_{Q \in \mathcal{CR}} \max_{a,b,c \in A} \min(Q(a,b), Q(b,c), Q(c,a)).$$

To obtain $\bar{\delta}_3$, it is therefore sufficient to consider reciprocal relations from $\mathcal{CR}$ with three elements and to find a set of values of $Q(a,b)$, $Q(b,c)$, $Q(c,a)$ such that $\min(Q(a,b), Q(b,c), Q(c,a))$ is maximal. From symmetry considerations it follows that the investigation may be restricted to reciprocal relations for which

$$Q(a,b) = Q(b,c) = Q(c,a) = q.$$

Expressing that such a reciprocal relation should be transitive with respect to the upper bound function (6.1), yields the condition $q^2 \leq 3q - 1 \leq 2q - q^2$, or equivalently $(3 - \sqrt{5})/2 \leq q \leq (\sqrt{5} - 1)/2$. As a consequence we have that

$$\delta_3 \leq \bar{\delta}_3 = (\sqrt{5} - 1)/2 \approx 0.618034.$$

Remark that this is precisely the golden section.

In general, for $m \geq 3$, let us introduce the following notations. We number the nodes of the complete graph from 1 to m and consider the cycle of length m in which the nodes appear in the natural order. The edges of this cycle are attributed equal weight a_1^m. We need to find the maximal value of a_1^m such that the reciprocal relation which underlies the graph is cycle-transitive with

respect to the upper bound function (6.1). Due to symmetry, we attribute to the edges starting at node i and ending at node $(i+k) \mod m$, irrespective of i, the same weight a_k^m, where $k \in \{1,2,\ldots,m-1\}$. Clearly, since the relation underlying this graph should be reciprocal, it holds that

$$a_k^m = 1 - a_{m-k}^m \quad \text{for all} \quad k \in \{1,2,\ldots,m-1\}.$$

We call these weighted graphs *max-optimal*.

In the first column of Table 6.6 the maximum values of a_1^m, which are precisely the minimum cutting levels $\bar{\delta}_m$, are listed for $m = 3,\ldots,7$. In the next two columns the values of a_k^m for $k = 2,\ldots,\lfloor m/2 \rfloor$ are shown. Remark that, because of the reciprocity, the remaining values, *i.e.*

$$a_k^m \quad \text{for} \quad k = \lfloor m/2 \rfloor + 1,\ldots,m,$$

can be found by complementation. Finally, in the last column, the polynomial equation whose largest real root provides the value of a_1^m is given for each m. Let us illustrate how this equation is found. For $m = 3$ there is only one cycle of length 3. By substituting $\alpha = \beta = \gamma = a_1^3$ in (6.3) the inequality $1 - a_1^3 \geq (a_1^3)^2$ is found. Maximizing a_1^3 in this equality yields the equation $(a_1^3)^2 + a_1^3 - 1 = 0$. Remark that no stricter condition is obtained by considering the inverse cycle and thus by expressing that $\alpha = \beta = \gamma = 1 - a_1^3$. Imposing that (6.3) holds for all cycles of length 3 for $m = 4$ yields the condition $1 - a_1^4 \geq a_1^3 a_2^4$. At the same time, because of symmetry, we know that $a_2^4 = 1 - a_2^4$. By maximizing a_1^4 the equation $3a_1^4 - 2 = 0$ is obtained. When $m = 5$, the following conditions are obtained:

$$\begin{cases} 1 - a_2^5 \geq a_1^5 a_2^5 \\ 1 - a_1^5 \geq a_1^5 (1 - a_2^5) \end{cases} \Leftrightarrow \begin{cases} a_1^5 \leq (1 - a_2^5)/a_2^5 \\ a_1^5 \leq 1/(2 - a_2^5) \end{cases}.$$

It is easily verified that a_1^5 attains a maximum at the intersection of the upper bounds on a_1^5, such that we obtain the equation

$$(1 - a_2^5)(2 - a_2^5) = a_2^5.$$

Substituting $a_2^5 = 1/(1 + a_1^5)$ into this equation yields $2(a_1^5)^2 - 1 = 0$. Analogous calculations can be made for $m \geq 6$.

A similar exercise can be made for the δ^*-transitivity in (5.34) which is stronger than our cycle-transitive formulation. If we denote by $\mathcal{D}R$ the set

m	$a_1^m = \bar{\delta}_m$	a_2^m	a_3^m	polynomial equation
3	0.61803			$(a_1^3)^2 + a_1^3 - 1 = 0$
4	0.66667	0.50000		$3a_1^4 - 2 = 0$
5	0.70711	0.58579		$2(a_1^5)^2 - 1 = 0$
6	0.72361	0.61803	0.50000	$5(a_1^6)^2 - 5a_1^6 + 1 = 0$
7	0.74227	0.65270	0.53209	$3(a_1^7)^3 - 3a_1^7 + 1 = 0$

Table 6.6: Weights of graphs with m nodes that represent max-optimal cycle-transitive reciprocal relations with respect to the upper bound function (6.1) whose strict cuts at $\bar{\delta}_m$ are free of cycles of length m.

of all finite reciprocal relations that exhibit δ^*-transitivity, it is clear that $\mathcal{MR} \subseteq \mathcal{DR} \subseteq \mathcal{CR}$. Let us now denote $\tilde{\delta}_m$ as the minimum cutting level for the class $\mathcal{DR}$. The results obtained by using a numerical solver for m up to 13 are summarized together with the values for $\bar{\delta}_m$ for m up to 13 in Table 6.7. Remark that the bounds obtained by the minimum cutting level $\tilde{\delta}_m$ for relations in $\mathcal{DR}$ are considerably sharper than those obtained by $\bar{\delta}_m$ for relations in $\mathcal{CR}$.

m	$\tilde{\delta}_m$	$\bar{\delta}_m$
3	0.555556	0.618034
4	0.585787	0.666667
5	0.605696	0.707107
6	0.620118	0.723607
7	0.631193	0.742227
8	0.640043	0.75
9	0.647324	0.767592
10	0.653448	0.773459
11	0.658690	0.780777
12	0.663243	0.788675
13	0.667243	0.792256

Table 6.7: Theoretical upper bounds on the minimum cutting level to avoid LEM cycles of length $l \leq m$ obtained by the minimum cutting level $\tilde{\delta}_m$ for reciprocal relations in $\mathcal{DR}$ and $\bar{\delta}_m$ for reciprocal relations in $\mathcal{CR}$.

It is known from the work of Yu [121] that the strict cut of any mutual rank probability relation at the value

$$\rho = \frac{1 + (\sqrt{2} - 1)\sqrt{2\sqrt{2} - 1}}{2} \approx 0.780048$$

yields a crisp relation that is transitive, and thus obviously is free of cycles of length m for any $m > 0$. Therefore, it must hold that

$$\lim_{m \to \infty} \delta_m \leq \rho \,.$$

6.2.2 Experimental results

Remark that the theoretical considerations on the minimum cutting levels concern the entire supersets $\mathcal{C}R$ and $\mathcal{D}R$ of $\mathcal{M}R$. Therefore the given bounds are not sharp when we restrict to posets of some given size. In this subsection we will experimentally compute the exact minimum cutting levels for posets on up to 13 elements.

An algorithm is introduced in Section 6.1.1 to retrieve all posets of a given size n with LEM cycles. It is straightforward to adapt this algorithm to keep track of the posets requiring the highest cutting level δ_m^n such that all mutual rank probabilities in a LEM cycle of length m are greater than or equal to δ_m^n. In Table 6.8, the minimum cutting levels to avoid cycles of length m in n-element posets are shown. Note that since no posets with $n \leq 13$ exist with LEM cycles of length 8, the cutting level for $m = 8$ is 0.5.

$n \backslash m$	3	4	5	6	7
9	**0.50314465**	0.5	0.5	0.5	0.5
10	**0.50396825**	**0.50284900**	0.5	0.5	0.5
11	**0.50619469**	0.50284900	0.5	0.5	0.5
12	**0.50735039**	**0.50866575**	**0.50039788**	**0.50242592**	0.5
13	**0.50886687**	0.50866575	**0.50289997**	**0.50246440**	**0.50018080**

Table 6.8: Minimum cutting level δ_m^n to avoid LEM cycles of length m in posets of size $n = 9, \ldots, 13$ for $m = 3, \ldots, 7$.

Since one can trivially establish a poset of size $n+1$ from a poset of size n with an equal minimum cutting level by adding an element which is either

smaller than, larger than or incomparable to all n elements, the minimum cutting levels δ_m^n are monotonically increasing for increasing n. Therefore, a minimum cutting level δ_m^n avoids all LEM cycles of length $l \leq m$. In Table 6.8 one can observe that for $n = 11$ no higher cutting level for avoiding cycles of length 4 is found than for $n = 10$ since $\delta_4^{11} = \delta_4^{10}$, and similarly it is found that $\delta_4^{13} = \delta_4^{12}$. Remark furthermore that a cutting level $\delta_m^n = 0.5$ indicates that no LEM cycles of length $l \leq m$ are present.

In Figures A.1-A.14 of Appendix A the posets requiring the non-trivial minimum cutting levels indicated in boldface in Table 6.8 are depicted by their Hasse diagrams. Note that the dual of a poset has an equal minimum cutting level, and therefore is not shown. However, four depicted posets are identical to their dual posets (Figures A.1, A.5, A.6 and A.14). It is also interesting to mention that some posets have multiple LEM cycles with an identical cutting level, while others have LEM cycles of different lengths. The poset of size 9 in Figure A.1, for example, has three cycles of length 3 with identical probabilities, while for the poset of size 12 in Figure A.5, aside from the cycle with length 3, a cycle of length 4 is present, since it holds that

$$\text{Prob}(5 > 7) = \text{Prob}(7 > 8) = \frac{6184}{12244}.$$

The poset of size 12 in Figure A.7 is quite remarkable in this respect, since aside from the cycle of length 5, cycles of length 3, 4 and 6 are present. The poset in Figure A.14 even has cycles of length 3, 4, 5, 6 and 7. Furthermore, the poset in Figure A.8 also has a cycle of length 3, the poset in Figure A.11 has cycles of length 3 and 4, and the posets in Figure A.12 and A.13 have both cycles of length 3. For some cutting levels multiple posets, aside from their dual versions, are found. This is the case for the posets of size 13 in Figure A.9 and A.10 which attain the minimum cutting level for cycles of length 3. In addition, the same is true for cycles of length 6 in Figure A.12 and A.13.

As an additional verification of the implementations of the algorithms, the mutual rank probabilities of all posets contained in this subsection have been verified using an implementation of an algorithm of Pruesse *et al.* [102] based on the fast generation of all linear extensions.

One of the aims of the experiments was to find common properties for posets with LEM cycles or to see a common structure emerging in the posets requiring the minimum cutting level. Indeed, if some common properties

are found it might be possible to confine the search space to one or more subclasses of posets, or at least to rule out several hopefully large enough subclasses. By doing this, one could hope to take a step further and to find all posets with LEM cycles for $n = 14$ or maybe even $n = 15$. However, to our surprise the posets have little in common. Possibly due to the fact that the posets are still very limited in size no common (sub)structures can yet be observed for increasing size. However, the symmetrical and relatively simple structure of the poset of size 12 in Figure A.6 requiring the minimum cutting level δ_4 inspired us in the next subsection to try to generalize it and to find a lower bound for δ_4 as sharp as possible for increasing poset size.

6.2.3 A lower bound for δ_4

Consider in Figure 6.4 the generalization of the poset in Figure A.6 requiring the highest cutting level to avoid cycles of length $l \leq 4$ in posets with 12 elements. Note that if we set $p = 1$ and $q = 0$ the original poset in Figure A.6 is obtained. We observed that by increasing q the cutting level needed to avoid cycles of size $l \leq 4$ increases as well. The same observation was made for increasing p. Moreover, there is exactly one cycle of length 4, consisting of the elements $\omega_1, \omega_2, \omega_3$ and ω_4, requiring this cutting level. Due to symmetry, the probabilities in the cycle are identical, *i.e.* $\text{Prob}(\omega_2 > \omega_1) = \text{Prob}(\omega_1 > \omega_4) = \text{Prob}(\omega_4 > \omega_3) = \text{Prob}(\omega_3 > \omega_2)$. Therefore it would be interesting to have an analytical expression for e.g. $\text{Prob}(\omega_2 > \omega_1)$ as a function of p and q, since we would then obtain cutting levels for removing cycles of length 4 in a family of posets which seem promising for obtaining a lower bound for δ_4 as close as possible to the theoretical upper bound $\tilde{\delta_4} \approx 0,585787$ from Table 6.7.

As a first step, we count the number of linear extensions of the poset, as this number will be the denominator of $\text{Prob}(\omega_2 > \omega_1)$. Note that the four indicated elements a, b, c and d on Figure 6.4 can occur in four different orders in a linear extension:

$$a < b < c < d,$$
$$a < b < d < c,$$
$$b < a < c < d,$$
$$b < a < d < c.$$

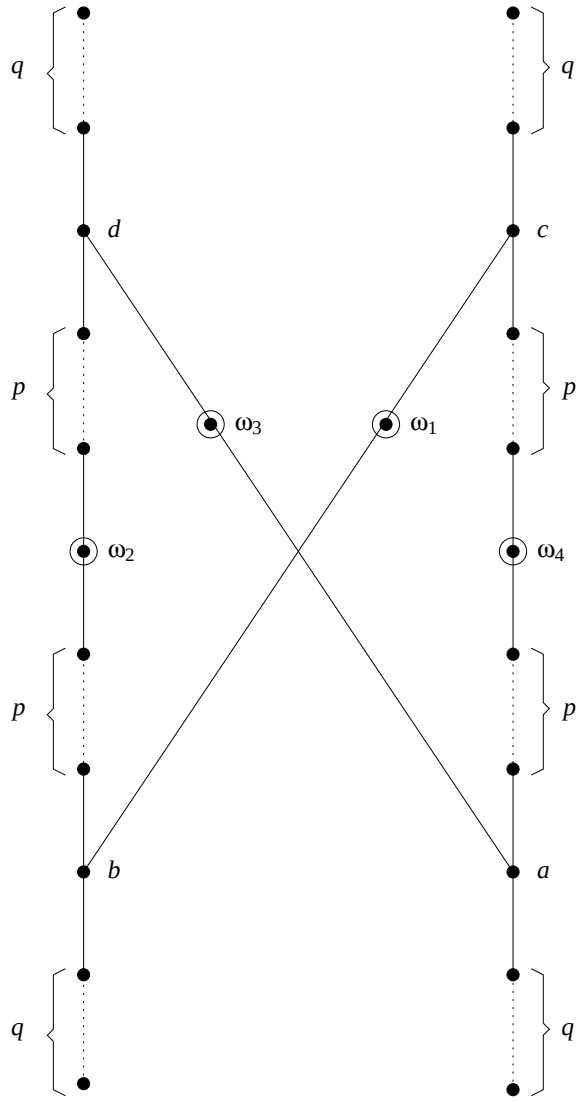


Figure 6.4: Generalization of the poset in Figure A.6 requiring the highest cutting level to avoid cycles of length $l \leq 4$ in posets with 12 elements.

Again due to symmetry, the number of linear extensions where $a < b$ is identical to those where $b < a$. We can therefore restrict to the orders $a < b < c < d$ and $a < b < d < c$ and multiply the expression found by 2 in order to obtain the total number of linear extensions.

Since we will often need the number of linear extensions of the poset consisting of two chains with lengths i and j, in what follows we will denote this number as $\kappa(i,j)$, *i.e.* we define

$$\kappa(i,j) = \binom{i+j}{j}.$$

The total number of linear extensions $N(p,q)$ can now be written as the following summation

$$\sum_{i_1=0}^{q} \kappa(q,i_1) \sum_{i_2=0}^{2p+1} \kappa(i_2, q-i_1) \left[N^{cd}(p,q,i_1,i_2) + N^{dc}(p,q,i_1,i_2) \right], \qquad (6.4)$$

where the two functions N^{cd} and N^{dc} cover the case that $c < d$ and $d < c$ respectively

$$N^{cd}(p,q,i_1,i_2) = \sum_{i_3=0}^{2p+1} \kappa(2p-i_2+1,i_3) \sum_{i_4=0}^{q} n^{cd} \kappa(i_4, 2p-i_3+1)\kappa(q-i_4,q),$$
$$N^{dc}(p,q,i_1,i_2) = \sum_{i_3=0}^{2p+1-i_2} \kappa(2p+1,i_3) \sum_{i_4=0}^{q} n^{dc} \kappa(i_4, 2p-i_2-i_3+1)\kappa(q-i_4,q).$$

We denote the number of elements in the poset that are smaller than ω_3 in each linear extension as $\underline{\omega}_3$. In other words, $\underline{\omega}_3$ is a lower bound for the position of ω_3. Similarly, we denote the upper bound on ω_3 when $c < d$ as $\overline{\omega}_3^{cd}$ and the upper bound on the position of ω_3 when $d < c$ as $\overline{\omega}_3^{dc}$. These bounds are given as follows:

$$\begin{aligned} \underline{\omega}_3 &= q + i_1 + 1, \\ \overline{\omega}_3^{cd} &= 4p + 2q + i_4 + 5, \\ \overline{\omega}_3^{dc} &= 2p + 2q + i_2 + i_3 + 3. \end{aligned}$$

Analogously we obtain a lower bound $\underline{\omega}_1$ and two upper bounds $\overline{\omega}_1^{cd}$ and

$\overline{\omega}_1^{dc}$ on the position of ω_1,

$$\begin{aligned}\underline{\omega}_1 &= 2q + i_2 + 2,\\ \overline{\omega}_1^{cd} &= 2p + 2q + i_3 + 3,\\ \overline{\omega}_1^{dc} &= 4p + 2q + i_4 + 5.\end{aligned}$$

Consider the case $c < d$. The element ω_1 can be freely inserted between positions $\underline{\omega}_1$ and $\overline{\omega}_1^{cd}$, and similarly the element ω_3 can be inserted between $\underline{\omega}_3$ and $\overline{\omega}_3^{cd}$. However, as can be seen in Figure 6.5, between $\underline{\omega}_1$ and $\overline{\omega}_1^{cd}$ an additional position for ω_3 appears due to the insertion of ω_1. In order to account for all possible positions of the two elements ω_1 and ω_3 in all linear extensions, we therefore have to add the term $\overline{\omega}_1^{cd} - \underline{\omega}_1 + 1$ as to obtain

$$n^{cd} = (\overline{\omega}_1^{cd} - \underline{\omega}_1 + 1)(\overline{\omega}_3^{cd} - \underline{\omega}_3 + 1) + \overline{\omega}_1^{cd} - \underline{\omega}_1 + 1. \tag{6.5}$$

When $d < c$ a similar argument holds as shown in Figure 6.6:

$$n^{dc} = (\overline{\omega}_1^{dc} - \underline{\omega}_1 + 1)(\overline{\omega}_3^{dc} - \underline{\omega}_3 + 1) + \overline{\omega}_3^{dc} - \underline{\omega}_1 + 1. \tag{6.6}$$

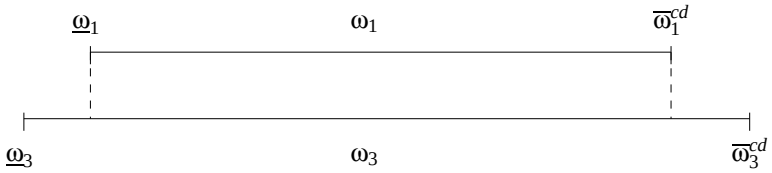


Figure 6.5: Lower and upper bounds on the positions of ω_1 and ω_3 when $c < d$.

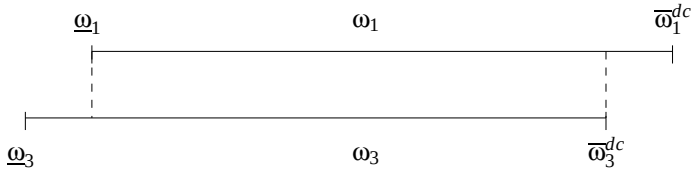


Figure 6.6: Lower and upper bounds on the positions of ω_1 and ω_3 when $d < c$.

After simplifying expressions (6.5) and (6.6) we obtain

$$n^{cd} = (2p - i_2 + i_3 + 2)(4p + q - i_1 + i_4 + 6),$$
$$n^{dc} = (4p - i_2 + i_4 + 4)(2p + q - i_1 + i_2 + i_3 + 3) + (2p + i_3 + 2).$$

It can be proven that the following equalities hold

$$\sum_{i_1=0}^{q} \kappa(q, i_1)\kappa(i_2, q - i_1) = \kappa(q + i_2 + 1, q),$$
$$\sum_{i_1=0}^{q} (q - i_1 + i_2 + 1)\kappa(q, i_1)\kappa(i_2, q - i_1) = (i_2 + 1)\kappa(q + i_2 + 2, q), \quad (6.7)$$

such that if we define $t = 2p + 1$, it is easily verified that $N(p, q)$ in expression (6.4) can be rewritten as

$$\begin{aligned}
&\sum_{i_2=0}^{t} \sum_{i_3=0}^{t} \kappa(t - i_2, i_3) \sum_{i_4=0}^{q} \kappa(i_4, t - i_3)\kappa(q - i_4, q) \cdot \\
&\qquad [(t - i_2 + i_3 + 1)(i_2 + 1)\kappa(q + i_2 + 2, q) + \\
&\qquad (t - i_2 + i_3 + 1)(2t - i_2 + i_4 + 3)\kappa(q + i_2 + 1, q)] \\
&+ \sum_{i_2=0}^{t} \sum_{i_3=0}^{t-i_2} \kappa(t, i_3) \sum_{i_4=0}^{q} \kappa(i_4, t - i_2 - i_3)\kappa(q - i_4, q) \cdot \\
&\qquad [(2t - i_2 + i_4 + 2)(i_2 + 1)\kappa(q + i_2 + 2, q) + \\
&\qquad (2t - i_2 + i_4 + 3)(t + i_3 + 1)\kappa(q + i_2 + 1, q)] . \quad (6.8)
\end{aligned}$$

In analogy to the equalities in (6.7) it is found that

$$\begin{aligned}
\sum_{i_4=0}^{q} \kappa(i_4, t - i_3)\kappa(q - i_4, q) &= \sum_{i_4=0}^{q} \kappa(q - i_4, t - i_3)\kappa(i_4, q) \\
&= \kappa(t + q - i_3 + 1, q),
\end{aligned}$$

and

$$\begin{aligned}
\sum_{i_4=0}^{q} \kappa(i_4, t - i_2 - i_3)\kappa(q - i_4, q) &= \sum_{i_4=0}^{q} \kappa(q - i_4, t - i_2 - i_3)\kappa(i_4, q) \\
&= \kappa(t + q - i_2 - i_3 + 1, q).
\end{aligned}$$

Moreover,

$$\begin{aligned}
&\sum_{i_4=0}^{q}(t-i_3+i_4+1)\kappa(i_4,t-i_3)\kappa(q-i_4,q)\\
&\quad=\sum_{i_4=0}^{q}(t+q-i_3-i_4+1)\kappa(q-i_4,t-i_3)\kappa(i_4,q)\\
&\quad=(t-i_3+1)\kappa(t+q-i_3+2,q),
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{i_4=0}^{q}(t-i_2-i_3+i_4+1)\kappa(i_4,t-i_2-i_3)\kappa(q-i_4,q)\\
&\quad=\sum_{i_4=0}^{q}(t+q-i_2-i_3-i_4+1)\kappa(q-i_4,t-i_2-i_3)\kappa(i_4,q)\\
&\quad=(t-i_3+1)\kappa(t+q-i_2-i_3+2,q),
\end{aligned}$$

such that (6.8) simplifies to

$$\begin{aligned}
&\sum_{i_2=0}^{t}\sum_{i_3=0}^{t}\kappa(t-i_2,i_3)\cdot\\
&\qquad[(t-i_2+i_3+1)(i_2+1)\kappa(t+q-i_3+1,q)\kappa(q+i_2+2,q)+\\
&\qquad(t-i_2+i_3+1)(t-i_3+1)\kappa(t+q-i_3+2,q)\kappa(q+i_2+1,q)+\\
&\qquad(t-i_2+i_3+1)(t-i_2+i_3+2)\kappa(t+q-i_3+1,q)\kappa(q+i_2+1,q)]\\
&+\sum_{i_2=0}^{t}\sum_{i_3=0}^{t-i_2}\kappa(t,i_3)\cdot\\
&\qquad[(t-i_2-i_3+1)(i_2+1)\kappa(t+q-i_2-i_3+2,q)\kappa(q+i_2+2,q)+\\
&\qquad(t+i_3+1)(i_2+1)\kappa(t+q-i_2-i_3+1,q)\kappa(q+i_2+2,q)+\\
&\qquad(t+i_3+1)(t-i_2-i_3+1)\kappa(t+q-i_2-i_3+2,q)\kappa(q+i_2+1,q)+\\
&\qquad(t+i_3+1)(t+i_3+2)\kappa(t+q-i_2-i_3+1,q)\kappa(q+i_2+1,q)]\,.
\end{aligned}\tag{6.9}$$

As a next step we calculate the number of linear extensions where $\omega_2 < \omega_1$, which is identical to the number of linear extensions of the poset in Figure 6.7. We will use an analogous technique, but since symmetry is lost in this case it is necessary to consider both cases $a < b$ and $b < a$.

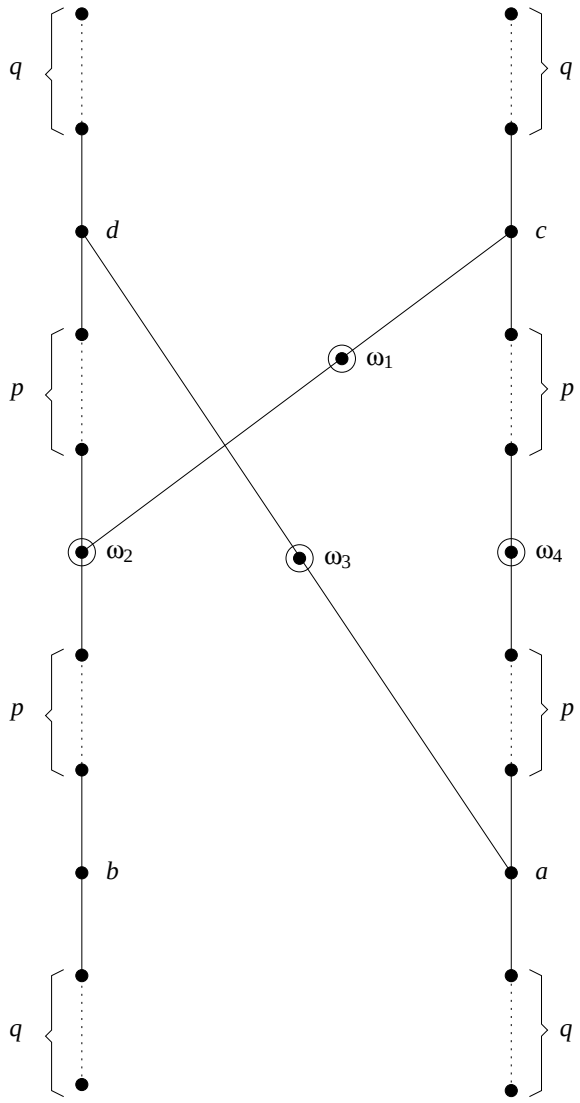


Figure 6.7: Generalization of the poset in Figure A.6 requiring the highest cutting level to avoid cycles of length $l \leq 4$ in posets with 12 elements where the ordered pair $\omega_2 < \omega_1$ is added.

We obtain for the number of linear extensions $M_{ab}(p,q)$ where $a < b$

$$\sum_{i_1=0}^{q+p+1} \kappa(q,i_1) \sum_{i_2=0}^{2p+1} \kappa(i_2, p+q-i_1+1) \left[M_{ab}^{cd}(p,q,i_1,i_2) + M_{ab}^{dc}(p,q,i_1,i_2) \right], \tag{6.10}$$

where the two functions M_{ab}^{cd} and M_{ab}^{dc} cover the case that $c < d$ and $d < c$ respectively:

$$\begin{aligned} M_{ab}^{cd}(p,q,i_1,i_2) &= \sum_{i_3=0}^{p} \kappa(2p-i_2+1,i_3) \sum_{i_4=0}^{q} m_{ab}^{cd} \kappa(i_4, p-i_3) \kappa(q-i_4,q), \\ M_{ab}^{dc}(p,q,i_1,i_2) &= \sum_{i_3=0}^{2p+1-i_2} \kappa(p,i_3) \sum_{i_4=0}^{q} m_{ab}^{dc} \kappa(2p-i_2-i_3+1,q) \kappa(q-i_4,q). \end{aligned}$$

The bounds on ω_1 and ω_3 are as follows

$$\begin{aligned} \underline{\omega}_3 &= q + i_1 + 1, \\ \overline{\omega}_3^{cd} &= 4p + 2q + i_4 + 5, \\ \overline{\omega}_3^{dc} &= 2p + 2q + i_2 + i_3 + 3, \\ \underline{\omega}_1 &= p + 2q + i_2 + 3, \\ \overline{\omega}_1^{cd} &= 3p + 2q + i_3 + 4, \\ \overline{\omega}_1^{dc} &= 4p + 2q + i_4 + 5. \end{aligned}$$

while

$$\begin{aligned} m_{ab}^{cd} &= (\overline{\omega}_1^{cd} - \underline{\omega}_1 + 1)(\overline{\omega}_3^{cd} - \underline{\omega}_3 + 1) + \overline{\omega}_1^{cd} - \underline{\omega}_1 + 1, \\ m_{ab}^{dc} &= (\overline{\omega}_1^{dc} - \underline{\omega}_1 + 1)(\overline{\omega}_3^{dc} - \underline{\omega}_3 + 1) + \overline{\omega}_3^{dc} - \underline{\omega}_1 + 1. \end{aligned}$$

and after simplification we obtain for $M_{ab}(p,q)$

$$\begin{aligned}
&\sum_{i_2=0}^{2p+1}\sum_{i_3=0}^{p}\kappa(2p-i_2+1,i_3)\,[(2p+i_3-i_2+2)\cdot\\
&\quad\{(i_2+1)\kappa(p+q-i_3+1,q)\kappa(q+i_2+2,p+q+1)+\\
&\quad(2p-i_2+i_3+3)\kappa(q+i_2+1,p+q+1)\kappa(p+q-i_3+1,q)+\\
&\quad(p-i_3+1)\kappa(q+i_2+1,p+q+1)\kappa(p+q-i_3+2,q)\}]\\
&+\sum_{i_2=0}^{2p+1}\sum_{i_3=0}^{2p+1-i_2}\kappa(p,i_3)\,[(p+i_3+1)\cdot\\
&\quad\{(2p-i_2-i_3+2)\kappa(q+i_2+1,p+q+1)\kappa(2p+q-i_2-i_3+3,q)+\\
&\quad(p+i_3+2)\kappa(q+i_2+1,p+q+1)\kappa(2p+q-i_2-i_3+2,q)\}\\
&\quad+(i_2+1)\cdot\\
&\quad\{(p+i_3+1)\kappa(q+i_2+2,p+q+1)\kappa(2p+q-i_2-i_3+2,q)+\\
&\quad(2m-i_2-i_3+2)\kappa(q+i_2+2,p+q+1)\kappa(2p+q-i_2-i_3+3,q)\}]\,.
\end{aligned}\tag{6.11}$$

Let us now consider the case $b<a$, *i.e.* $M_{ba}(p,q)$,

$$\sum_{i_1=0}^{q}\kappa(p+q+1,i_1)\sum_{i_2=0}^{p}\kappa(i_2,q-i_1)\left[M_{ba}^{cd}(p,q,i_1,i_2)+M_{ba}^{dc}(p,q,i_1,i_2)\right],\tag{6.12}$$

where the two functions M_{ba}^{cd} and M_{ba}^{dc} cover the case that $c<d$ and $d<c$ respectively,

$$\begin{aligned}
M_{ba}^{cd}(p,q,i_1,i_2)&=\sum_{i_3=0}^{2p+1}\kappa(p-i_2,i_3)\sum_{i_4=0}^{q}m_{ba}^{cd}\kappa(i_4,2p-i_3+1)\kappa(q-i_4,q),\\
M_{ba}^{dc}(p,q,i_1,i_2)&=\sum_{i_3=0}^{p-i_2}\kappa(2p+1,i_3)\sum_{i_4=0}^{q}m_{ba}^{dc}\kappa(p-i_2-i_3,q)\kappa(q-i_4,q).
\end{aligned}$$

The bounds on ω_1 and ω_3 are as follows

$$\begin{aligned}
\underline{\omega}_3 &= p+q+i_1+2,\\
\overline{\omega}_3^{cd} &= 4p+2q+i_4+5,\\
\overline{\omega}_3^{dc} &= 3p+2q+i_2+i_3+4,\\
\underline{\omega}_1 &= p+2q+i_2+3,\\
\overline{\omega}_1^{cd} &= 2p+2q+i_3+3,\\
\overline{\omega}_1^{dc} &= 4p+2q+i_4+5.
\end{aligned}$$

while

$$\begin{aligned}
m_{ba}^{cd} &= (\overline{\omega}_1^{cd}-\underline{\omega}_1+1)(\overline{\omega}_3^{cd}-\underline{\omega}_3+1)+\overline{\omega}_1^{cd}-\underline{\omega}_1+1.\\
m_{ba}^{dc} &= (\overline{\omega}_1^{dc}-\underline{\omega}_1+1)(\overline{\omega}_3^{dc}-\underline{\omega}_3+1)+\overline{\omega}_3^{dc}-\underline{\omega}_1+1,
\end{aligned}$$

and after simplification we obtain for $M_{ba}(p,q)$

$$\begin{aligned}
&\sum_{i_2=0}^{p}\sum_{i_3=0}^{2p+1}\kappa(p-i_2,i_3)\,[(p-i_2+i_3+1)\cdot\\
&\quad\{(2p-i_3+2)\kappa(p+q+i_2+2,q)\kappa(2p+q-i_3+3,q)+\\
&\quad(p-i_2+i_3+2)\kappa(p+q+i_2+2,q)\kappa(2p+q-i_3+2,q)+\\
&\quad(i_2+1)\kappa(p+q+i_2+3,q)\kappa(2p+q-i_3+2,q)\}]\\
&+\sum_{i_2=0}^{p}\sum_{i_3=0}^{p-i_2}\kappa(2p+1,i_3)\,[(2p+i_3+2)\cdot\\
&\quad\{(p-i_2-i_3+1)\kappa(p+q-i_2-i_3+2,q)\kappa(p+q+i_2+2,q)+\\
&\quad(2p+i_3+3)\kappa(p+q-i_2-i_3+1,q)\kappa(p+q+i_2+2,q)\}\\
&\quad+(i_2+1)\cdot\\
&\quad\{(p-i_2-i_3+1)\kappa(p+q-i_2-i_3+2,q)\kappa(p+q+i_2+3,q)+\\
&\quad(2p+i_3+2)\kappa(p+q-i_2-i_3+1,q)\kappa(p+q+i_2+3,q)\}]\,.
\end{aligned}\tag{6.13}$$

It is clear that for arbitrary p and q the mutual rank probability $\mathrm{Prob}(\omega_2 > \omega_1)$ is now given by the expression

$$1 - \frac{M_{ab}(p,q) + M_{ba}(p,q)}{2N(p,q)}. \tag{6.14}$$

We will now consider the case where $q \to \infty$. We remark that for the functions in expressions (6.9), (6.11) and (6.13) having the form

$$\kappa(q+i, q+j) = \binom{2q+i+j}{q+j} = \frac{(2q+i+j)!}{(q+i)! \cdot (q+j)!}$$

Stirling's approximation can be used, *i.e.*

$$f(q)! \approx \sqrt{2\pi \cdot f(q)} \cdot f(q)^{f(q)} \cdot e^{-f(q)} \quad \text{when } q \to \infty,$$

such that

$$\kappa(q+i, q+j) \approx \frac{2^{2q+i+j}}{\sqrt{\pi \cdot n}} \quad \text{when } q \to \infty.$$

Due to the nature of the fraction in expression (6.14), it is equivalent to substitute

$$\kappa(q+i, q+j) \quad \text{by} \quad 2^{i+j}. \tag{6.15}$$

It is now feasible to compute $\text{Prob}(\omega_2 > \omega_1)$ for given p when $q \to \infty$. Some values are given in Table 6.9 and a plot is shown in Figure 6.8.

As can be seen, the minimum cutting level quickly increases for increasing values of p, but soon the rate at which the function increases diminishes to attain values slightly below $0,58$. Recall that the upper bound $\tilde{\delta}_4$ on δ_4 is approximately $0,585787$, such that we obtain a quite narrow interval for the minimum cutting level δ_4. It comes as no surprise that we do not attain $\tilde{\delta}_4$ since, as already mentioned, this upper bound is obtained by a generalization of the mutual rank probability relation. Moreover, it can be expected that posets with more than 12 elements that do not fall into this family of posets provide sharper lower bounds.

p	$\text{Prob}(\omega_2 > \omega_1)$	
1	8/15	≈ 0.533333
20	419/731	≈ 0.573187
40	1051/1826	≈ 0.575575
60	1158/2009	≈ 0.576406
80	12223/21190	≈ 0.576829
100	3163/5481	≈ 0.577084
120	9071/15714	≈ 0.577256
140	18464/31979	≈ 0.577379
160	16041/27778	≈ 0.577471
180	10133/17545	≈ 0.577543
200	74953/129766	≈ 0.577601

Table 6.9: The mutual rank probabilities $\text{Prob}(\omega_2 > \omega_1)$ for given p when $q \to \infty$.

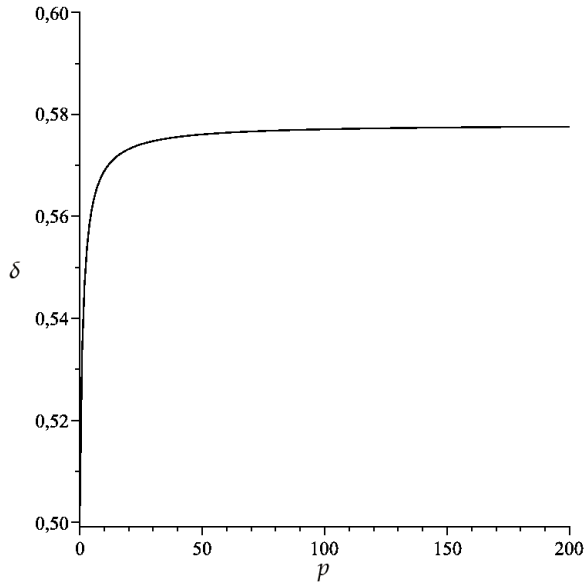
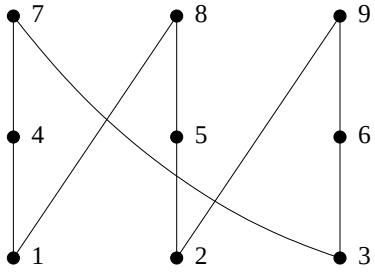


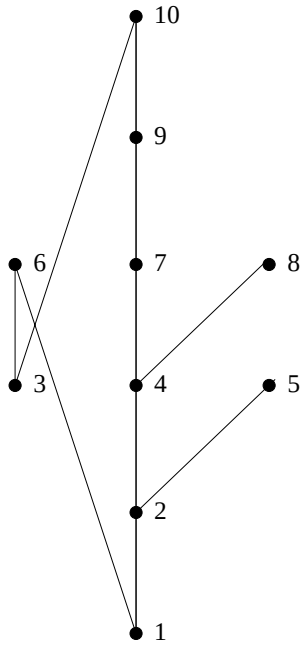
Figure 6.8: A plot of the mutual rank probabilities $\text{Prob}(\omega_2 > \omega_1)$ for $q \to \infty$.

A Posets requiring minimum cutting levels δ_m^n



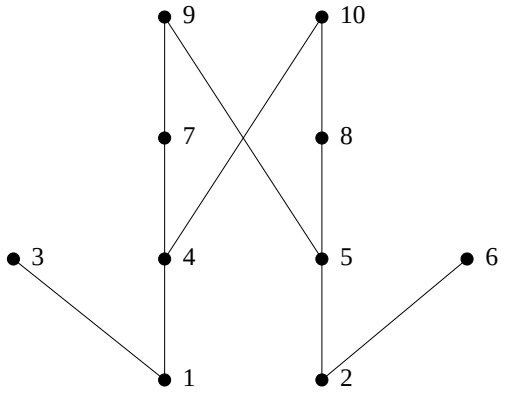
$$\text{Prob}(7 > 8) = \text{Prob}(8 > 9) = \text{Prob}(9 > 7) = \frac{720}{1431} \approx 0,50314465$$
$$\text{Prob}(4 > 5) = \text{Prob}(5 > 6) = \text{Prob}(6 > 4) = \frac{720}{1431} \approx 0,50314465$$
$$\text{Prob}(1 > 2) = \text{Prob}(2 > 3) = \text{Prob}(3 > 1) = \frac{720}{1431} \approx 0,50314465$$

Figure A.1: Poset of size 9 with a LEM cycle requiring the minimal cutting level δ_3^9.



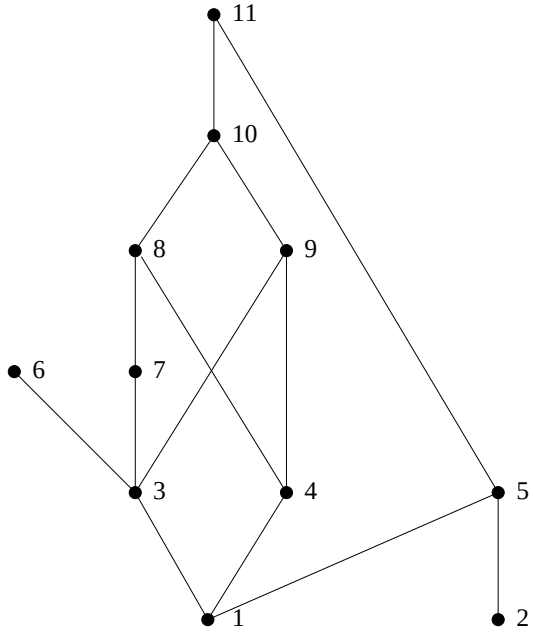
$$\text{Prob}(8 > 6) = \text{Prob}(6 > 9) = \frac{508}{1008} \approx 0,50396825$$
$$\text{Prob}(9 > 8) = \frac{512}{1008}$$

Figure A.2: Poset of size 10 with a LEM cycle requiring the minimal cutting level δ_3^{10}.



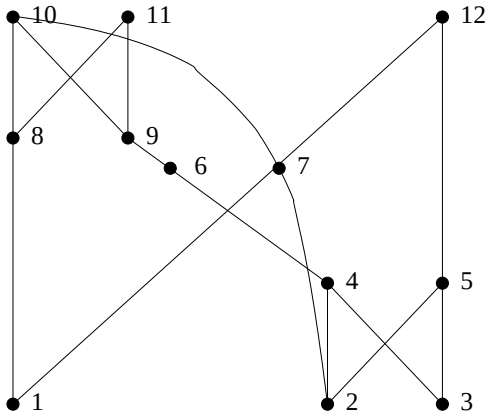
$$\text{Prob}(7 > 3) = \text{Prob}(3 > 8) = \text{Prob}(8 > 6) = \text{Prob}(6 > 7) = \frac{1765}{3510} \approx 0,50284900$$

Figure A.3: Poset of size 10 with a LEM cycle requiring the minimal cutting level δ_4^{10}.



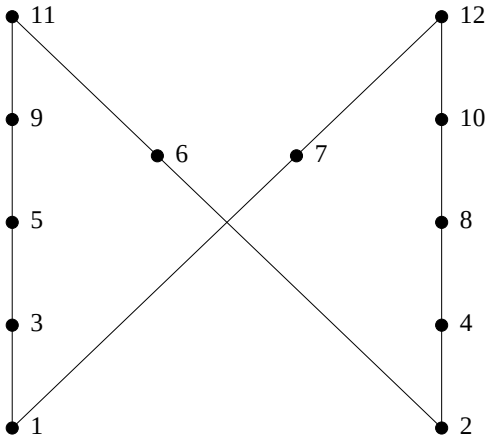
$$\text{Prob}(5 > 8) = \frac{1146}{2260}$$
$$\text{Prob}(8 > 6) = \frac{1144}{2260} \approx 0,50619469$$
$$\text{Prob}(6 > 5) = \frac{1145}{2260}$$

Figure A.4: Poset of size 11 with a LEM cycle requiring the minimal cutting level δ_3^{11}.



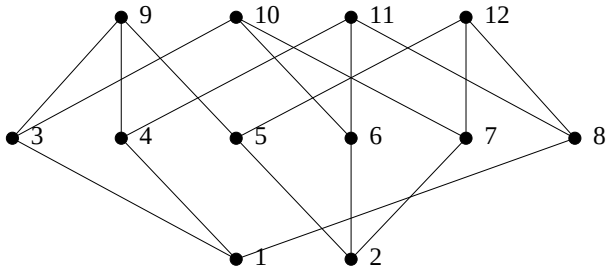
$$\text{Prob}(8 > 6) = \text{Prob}(6 > 5) = \frac{6214}{12244}$$
$$\text{Prob}(5 > 8) = \frac{6212}{12244} \approx 0,50735039$$

Figure A.5: Poset of size 12 with a LEM cycle requiring the minimal cutting level δ_3^{12}.



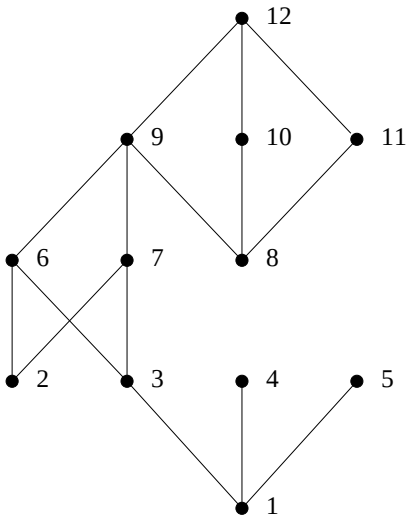
$$\text{Prob}(5 > 7) = \text{Prob}(7 > 8) = \text{Prob}(8 > 6) = \text{Prob}(6 > 5) = \frac{7396}{14540} \approx 0,50866575$$

Figure A.6: Poset of size 12 with a LEM cycle requiring the minimal cutting level δ_4^{12}.



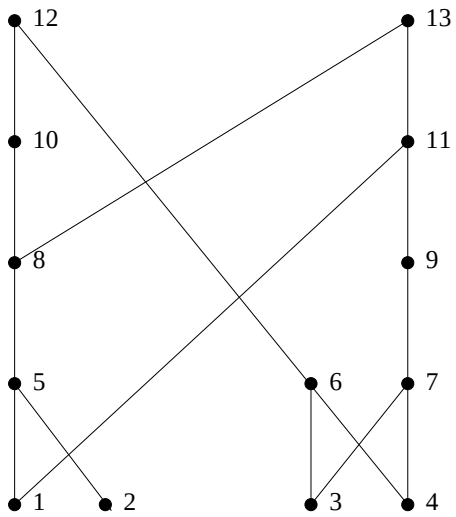
$$\text{Prob}(5 > 4) = \text{Prob}(4 > 3) = \frac{60400}{120640}$$
$$\text{Prob}(3 > 6) = \text{Prob}(6 > 8) = \text{Prob}(8 > 5) = \frac{60368}{120640} \approx 0,50039788$$

Figure A.7: Poset of size 12 with a LEM cycle requiring the minimal cutting level δ_5^{12}.



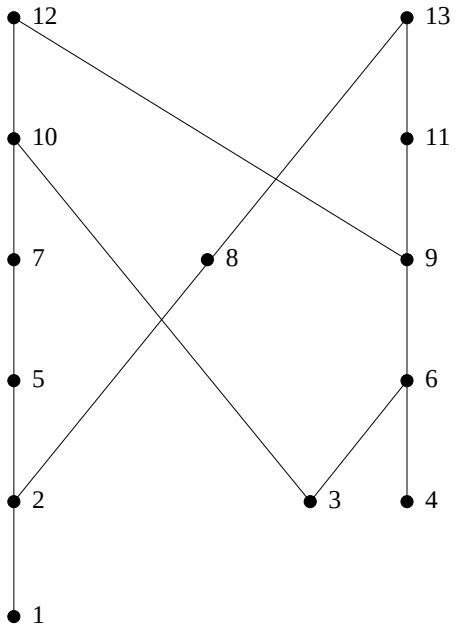
$$\begin{aligned}
\text{Prob}(7 > 4) = \text{Prob}(6 > 5) &= \frac{46392}{92336} \approx 0,50242592 \\
\text{Prob}(4 > 10) = \text{Prob}(5 > 11) &= \frac{46560}{92336} \\
\text{Prob}(10 > 6) = \text{Prob}(11 > 7) &= \frac{46850}{92336}
\end{aligned}$$

Figure A.8: Poset of size 12 with a LEM cycle requiring the minimal cutting level δ_6^{12}.



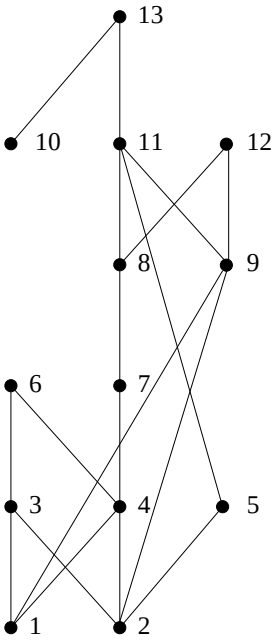
$$\text{Prob}(6 > 8) = \frac{12240}{24022}$$
$$\text{Prob}(8 > 9) = \frac{12262}{24022}$$
$$\text{Prob}(9 > 6) = \frac{12224}{24022} \approx 0,50886687$$

Figure A.9: First poset of size 13 with a LEM cycle requiring the minimal cutting level δ_3^{13}.



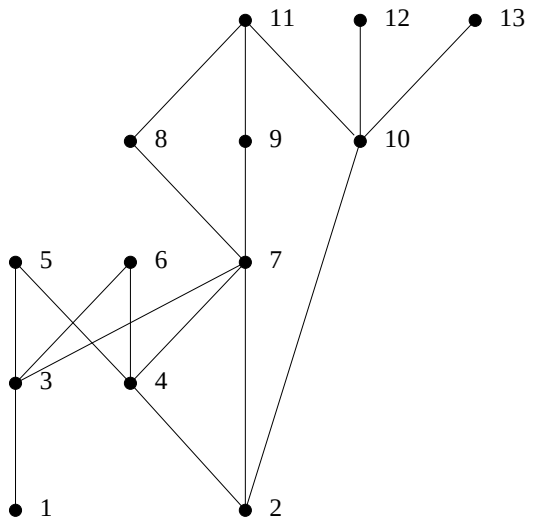
$$\text{Prob}(7 > 8) = \frac{6112}{12011} \approx 0,50886687$$
$$\text{Prob}(8 > 9) = \frac{6120}{12011}$$
$$\text{Prob}(9 > 7) = \frac{6131}{12011}$$

Figure A.10: Second poset of size 13 with a LEM cycle requiring the minimal cutting level δ_3^{13}.



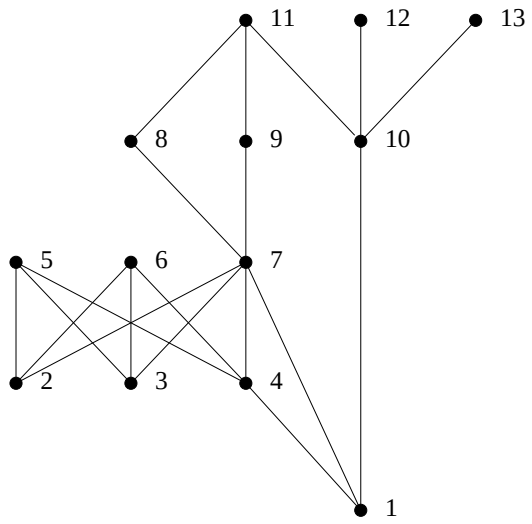
$$\begin{aligned}
\text{Prob}(10 > 9) &= \frac{33871}{67242} \\
\text{Prob}(9 > 5) &= \frac{33916}{67242} \\
\text{Prob}(5 > 7) &= \frac{33816}{67242} \approx 0,50289997 \\
\text{Prob}(7 > 3) &= \frac{33834}{67242} \\
\text{Prob}(3 > 10) &= \frac{34151}{67242}
\end{aligned}$$

Figure A.11: Poset of size 13 with a LEM cycle requiring the minimal cutting level δ_5^{13}.



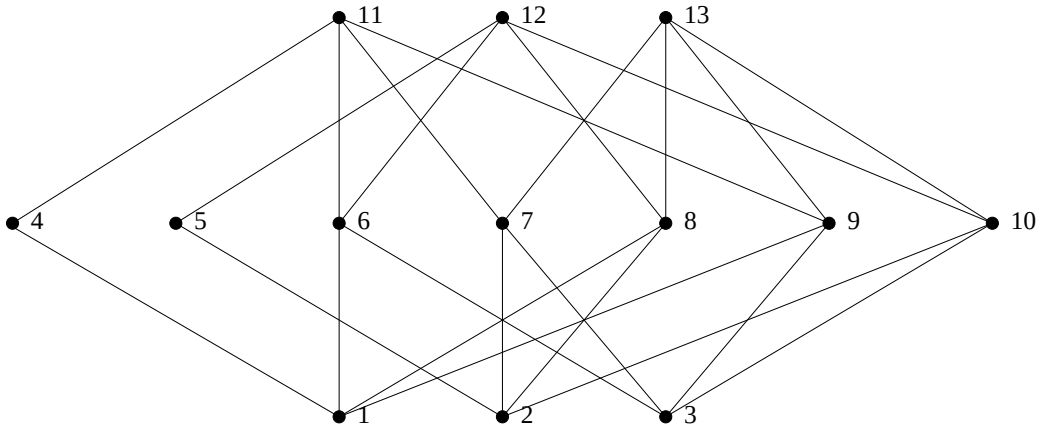
$$\begin{aligned}
\text{Prob}(12 > 9) = \text{Prob}(13 > 8) &= \frac{66354}{131472} \\
\text{Prob}(9 > 6) = \text{Prob}(8 > 5) &= \frac{66060}{131472} \approx 0,50246440 \\
\text{Prob}(6 > 13) = \text{Prob}(5 > 12) &= \frac{66306}{131472}
\end{aligned}$$

Figure A.12: First poset of size 13 with a LEM cycle requiring the minimal cutting level δ_6^{13}.



$$\begin{aligned}
\text{Prob}(12 > 9) = \text{Prob}(13 > 8) &= \frac{132708}{262944} \\
\text{Prob}(9 > 6) = \text{Prob}(8 > 5) &= \frac{132120}{262944} \approx 0,50246440 \\
\text{Prob}(6 > 13) = \text{Prob}(5 > 12) &= \frac{132612}{262944}
\end{aligned}$$

Figure A.13: Second poset of size 13 with a LEM cycle requiring the minimal cutting level δ_6^{13}.



$$\begin{aligned}\text{Prob}(11 > 6) = \text{Prob}(6 > 8) = \text{Prob}(4 > 11) &= \frac{268352}{536510} \approx 0,50018080\\ \text{Prob}(8 > 10) = \text{Prob}(5 > 4) &= \frac{268384}{536510}\\ \text{Prob}(10 > 12) = \text{Prob}(12 > 5) &= \frac{268465}{536510}\end{aligned}$$

Figure A.14: Poset of size 13 with a LEM cycle requiring the minimal cutting level δ_7^{13}.

Conclusion

As the title of this work indicates, the central theme in this work is the computation of rank probabilities of posets. Since the probability space consists of the set of all linear extensions of a given poset equipped with the uniform probability measure, in first instance we developed algorithms to explore this probability space efficiently. We considered in particular the problem of counting the number of linear extensions and the ability to generate extensions uniformly at random. Algorithms based on the lattice of ideals representation of a poset were developed.

Since a weak order extension of a poset can be regarded as an order on the equivalence classes of a partition of the given poset not contradicting the underlying order, and thus as a generalization of the concept of a linear extension, algorithms were developed to count and generate weak order extensions uniformly at random as well. However, in order to reduce the inherent complexity of the problem, the cardinalities of the equivalence classes is fixed a priori. Due to the exponential nature of these algorithms this approach is still not always feasible, forcing one to resort to approximative algorithms if this is the case. It is well known that Markov chain Monte Carlo methods can be used to generate linear extensions uniformly at random, but no such approaches have been used to generate weak order extensions. Therefore, an algorithm that can be used to sample weak order extensions uniformly at random was developed.

A monotone assignment of labels to objects from a poset corresponds to the choice of a weak order extension of the poset. Since the random monotone assignment of such labels is a step in the generation process of random

monotone data sets, the ability to generate random weak order extensions clearly is of great importance. The contributions from this part therefore prove useful in e.g. the field of supervised classification, where a need for synthetic random monotone data sets is present.

The second part focused on the ranking of the elements of a partially ordered set. Algorithms for the computation of the (mutual) rank probabilities that avoid having to enumerate all linear extensions were suggested and applied to a real-world data set containing pollution data of several regions in Baden-Württemberg (Germany). With the emergence of several initiatives aimed at protecting the environment like the REACH (Registration, Evaluation, Authorisation and Restriction of Chemicals) project of the European Union, the need for objective methods to rank chemicals, regions, etc. on the basis of several criteria still increases. Additionally, an interesting relation between the mutual rank probabilities and the average rank probabilities is proven.

The third and last part studied the transitivity properties of the mutual rank probabilities and the closely related linear extension majority cycles or LEM cycles for short. The type of transitivity was translated into the cycle-transitivity framework, which has been tailor-made for characterizing transitivity of reciprocal relations, and has been proved to be situated between strong stochastic transitivity and a new type of transitivity called δ^*-transitivity. It is shown that the latter type is situated between partial stochastic transitivity and a kind of product transitivity. Furthermore, theoretical upper bounds for the minimum cutting level to avoid LEM cycles were found. Cutting levels for posets on up to 13 elements were obtained experimentally and a theoretic lower bound for the cutting level to avoid LEM cycles of length 4 has been computed. Especially in this part of the work there is still a lot of research to be done. The type of transitivity exhibited by the mutual rank probabilities is far from characterized and it is not yet understood when precisely LEM cycles occur.

The research presented in this work has been published in international peer-reviewed journals ([36, 37, 39, 40, 41, 42, 43, 44]) and has been presented on international conferences. A Java implementation of several of the algorithms presented in this work, as well as binary files containing all posets on up to 13 elements with LEM cycles, can be downloaded from the website `http://www.kermit.ugent.be`.

Nederlandstalige samenvatting

Het vergelijken van objecten en maken van een rangschikking ervan op basis van criteria is een vaak terugkomende taak. Denk bijvoorbeeld aan het rangschikken van verschillende sollicitanten voor een openstaande betrekking, het rangschikken van regio's volgens vervuiling door schadelijke chemische verbindingen, het rangschikken van bedrijven volgens kredietwaardigheid, enz. In dit werk beperken we ons tot monotone rangschikkingen: wanneer een object x voor alle criteria minstens even goede scores behaalt als een object y zal object x nooit lager gerangschikt worden dan object y.

Beschouw bij het rangschikken van regio's volgens hun vervuiling de situatie waarbij een bepaalde regio x gekenmerkt wordt door een sterkere vervuiling van lood dan een regio y, terwijl terzelfdertijd regio x gekenmerkt wordt door een minder sterke vervuiling van cadmium dan regio y. Op basis van deze twee criteria kunnen beide regio's niet vergeleken worden zolang geen gemeenschappelijke schaal voor de criteria wordt toegekend. Het toekennen van een dergelijke schaal komt in feite neer op het toekennen van gewichten aan ieder criterium en is dikwijls controversieel omwille van het subjectieve karakter. Precies omwille van deze reden wordt ervoor gekozen om niet langer te eisen dat alle objecten vergelijkbaar zijn. Omdat objecten nu onderling onvergelijkbaar kunnen zijn wordt de verzameling van objecten een partieel geordende verzameling of poset genoemd.

In het tweede hoofdstuk wordt een formele introductie gegeven op het rangschikken en worden de nodige begrippen omtrent partieel geordende verza-

melingen aangebracht. Er wordt aangetoond dat het rangschikken van de elementen van een partieel geordende verzameling neerkomt op het bepalen van een lineaire extensie. Verder wordt ingegaan op een alternatieve representatie van een partieel geordende verzameling door zijn tralie van idealen. Tenslotte wordt het algoritme van Habib *et al.* [70] voor het opbouwen van deze tralie, dat tot op een constante factor na optimaal is, besproken. Een lemma van Bonnet *et al.* [10] geeft immers een bijectief verband tussen paden van maximale lengte in deze tralie van idealen en de lineaire extensies van de partieel geordende verzameling zelf.

Een belangrijke tak binnen het machinaal leren is het gesuperviseerd classificeren. In een classificatieprobleem wil men labels uit een lineair geordende verzameling toekennen aan objecten uit een partieel geordende verzameling. Merk op dat rangschikken een speciaal geval is van classificeren: de labelverzameling is de verzameling van natuurlijke getallen $\{1, 2, \ldots, n\}$ en de unieke labels corresponderen met de posities van de objecten $\{x_1, x_2, \ldots, x_n\}$ in de rangschikking. Met het predicaat gesuperviseerd doelt men op het aanwezig zijn van een dataverzameling met leervoorbeelden, *i.e.* objecten met een reeds toegekend label. Op basis van deze leervoorbeelden is het de bedoeling nieuwe objecten die nog niet aanwezig zijn in de dataverzameling een label toe te kennen. Dataverzamelingen uit de praktijk bevatten dikwijls ruis en zijn dus niet noodzakelijk monotoon: het is mogelijk dat een object x dat op alle criteria minstens even goed scoort als object y en voor één of meerdere criteria zelfs beter scoort, toch een slechter label krijgt toegekend. Veel algoritmen voor het gesuperviseerd classificeren kunnen echter slechts met monotone dataverzamelingen omgaan. Wil men nu op een objectieve wijze de performantie van verschillende gesuperviseerde classificatiealgoritmen kunnen vergelijken dan heeft men nood aan synthetisch geproduceerde monotone dataverzamelingen, waarbij het wenselijk is dat iedere dataverzameling met eenzelfde probabiliteit gegenereerd wordt. In het derde hoofdstuk wordt voor dit probleem een algoritme ontworpen. Een dataverzameling bestaat uit een verzameling van objecten en labels, samen met een monotone toekenning van labels aan de objecten. De partieel geordende verzameling van objecten en de verzameling van labels kunnen worden gegenereerd volgens a priori gekozen distributiefuncties. Wanneer het aantal objecten en het aantal labels identiek is, komt het genereren van een motonone toekenning neer op het genereren van een lineaire extensie van de partieel geordende verzameling van objecten. In het algemene geval, waarbij het aantal labels niet vastgelegd wordt, dient een zogenaamde zwak geordende extensie van de verzameling

te worden gegenereerd. Zwak geordende extensies kunnen gezien worden als lineaire extensies waarbij zich mogelijke ex-aequo's kunnen voordoen en dus equivalentieklassen aanwezig zijn. Objecten die zich in dezelfde equivalentieklasse bevinden worden dan eenzelfde label toegekend. Om de complexiteit van het probleem te reduceren wordt een algoritme ontworpen waarbij de cardinaliteiten van deze equivalentieklassen op voorhand worden vastgelegd. De inherente exponentiële complexiteit laat echter niet toe dit algoritme in te zetten voor grotere verzamelingen. Precies omwille van deze beperking wordt verder aandacht besteed aan benaderende algoritmen die dergelijke dataverzamelingen genereren op een (bijna) uniforme wijze.

In het vierde hoofdstuk worden de (mutuele) rangschikkingsprobabiliteiten van de elementen van een partieel geordende verzameling aangebracht. De rangschikkingsprobabiliteit $\mathrm{Prob}(\mathrm{rank}(x) = i)$ dat element x op positie i gerangschikt wordt is gedefinieerd als de fractie van de lineaire extensies waarbij x op positie i voorkomt. De mutuele rangschikkingsprobabilteit $\mathrm{Prob}(x > y)$ voor twee elementen x en y is gelijk aan de fractie van de lineare extensies waarin x voor y gerangschikt wordt. Opnieuw worden algoritmen geïntroduceerd die gebaseerd zijn op de voorstelling van een partieel geordende verzameling als tralie van idealen voor het berekenen van deze probabiliteiten. Er wordt tevens een interessant verband tussen de mutuele rangschikkingsprobabiliteiten en de gemiddelde rangschikkingsprobabiliteiten bewezen. Tenslotte worden de algoritmen toegepast op een dataverzameling uit de praktijk van concentraties van lood, cadmium, zink en zwavel in de kruidlaag van 59 regios in Baden-Württemberg (Duitsland). Er wordt uiteengezet hoe deze probabiliteiten kunnen helpen bij het bekomen van een rangschikking van de regio's volgens vervuiling.

In het volgende hoofdstuk worden de transitiviteitseigenschappen van de mutuele rangschikkingsprobabiliteitenrelatie bestudeerd. Transitiviteit is een eenvoudige maar krachtige eigenschap van relaties die opgelegd wordt aan ieder triplet van elementen. Fishburn [60] introduceerde het begrip proportionele transitiviteit met als doel het zo precies mogelijk karakteriseren van de mutuele rangschikkingsprobabiliteitenrelatie. We trachten deze vorm van transitiviteit te bestuderen binnen het kader van cykeltransitiviteit, specifiek ontworpen voor het karakteriseren van de transitiviteit van reciproke relaties. Kenmerkend voor deze cykeltransitiviteit is het cyclische karakter waarmee de tripletten doorlopen worden. Er wordt aangetoond dat het transitiviteitstype vertoond door de mutuele rangschikkingsproba-

biliteitenrelatie te situeren is tussen sterk stochastische transitiviteit en een nieuw type transitiviteit, genaamd δ^*-transitiviteit. Dit laatste type is zelf gesitueerd tussen partiële stochastische transitiviteit en een tussenvorm van producttransitiviteit.

Een direct gevolg van het feit dat de mutuele rangschikkingsprobabiliteitenrelatie niet partieel stochastisch transitief is, is het voorkomen van zogenaamde lineaire extensie majoriteiten cykels of kortweg LEM cykels. Een LEM cykel komt voor wanneer de mutuele rangschikkingsprobabiliteiten in een cyclisch doorlopen triplet allen strikt groter zijn dan 0.5. Gehrlein *et al.* [69] toonden aan dat dergelijke cykels enkel voorkomen bij partieel geordende verzamelingen met minstens 9 elementen. In de literatuur werden schattingen gemaakt voor de waarschijnlijkheid om LEM cykels aan te treffen in posets tot 12 elementen. Gebaseerd op de algoritmen ontwikkeld in de vorige hoofdstukken trachten we echter in het laatste hoofdstuk alle (niet-isomorfe) partieel geordende verzamelingen tot en met 13 elementen die één of meer LEM cykels bevatten te tellen en op te slaan in een databank voor latere referentie. Typisch is de geheugengrootte de beperkende factor bij deze aanpak gebaseerd op de tralie van idealen, vermits de tralie in zijn totaliteit in het geheugen geplaatst dient te worden. Hoewel het aantal idealen nog steeds exponentieel is in het aantal elementen, hebben in dit geval de verzamelingen nooit een cardinaliteit groter dan 13, waardoor deze aanpak adequaat is. De bekomen resultaten bevestigen een conjectuur geformuleerd door Gehrlein *et al.* [69] waarin gesteld wordt dat de probabiliteit om in een willekeurige partieel geordende verzameling een LEM cykel te vinden stijgt voor stijgende cardinaliteit. Echter, de mate waarin de probabiliteit toeneemt lijkt af te nemen voor stijgende cardinaliteit.

In een tweede gedeelte van het zesde hoofdstuk wordt onderzoek verricht naar de zogenaamde minimale snijniveau's voor het vermijden van LEM cykels van opgegeven lengte. Dit minimale snijniveau δ_m is het kleinst mogelijke getal zodat tenminste één mutuele rangschikkingsprobabiliteit in een arbitraire LEM cykel van lengte maximaal m kleiner dan of gelijk aan δ_m is. In eerste instantie worden theoretische bovengrenzen bepaald voor δ_m op basis van de gekende transitiviteitseigenschappen. Vervolgens worden experimenteel de minimale snijniveau's bepaald voor partieel geordende verzamelingen tot 13 elementen. Tenslotte wordt, geïnspireerd op de experimenteel gevonden partieel geordende verzameling met 12 elementen die aanleiding geeft tot het hoogste vereiste snijniveau voor cykels van lengte 4, een theoretische ondergrens voor δ_4 bekomen.

Index

1-Lipschitz property, 85
1/3-2/3-conjecture, 120
2-increasing, 85
$B(i)$, 19
C, 10
M_P, 100
T, 19
T-transitivity, 87
Child, 20
Class, 49
$\mathcal{E}$, 12
$\mathcal{I}$, 14
$\mathcal{I}_f$, 34
ImPred, 19
ImSucc, 19
$\bowtie$, 20
LEF, 34
LEI, 63
Label, 20, 49
Parent, 20
Paths, 49, 51
Pred, 19
Succ, 19
$T_{\mathbf{D}}$, 85
$(T_\lambda^{\mathbf{F}})_{\lambda\in[0,\infty]}$, 85
$T_{\mathbf{L}}$-transitivity, 95
$T_{\mathbf{M}}$, 85
$T_{\mathbf{M}}$-transitivity, 95
$T_{\mathbf{P}}$, 85
$T_{\mathbf{P}}$-transitivity, 96
Visited, 34, 49
VisitedElement, 68
δ_m, 122
$\downarrow$, 14
$\prec$, 10
$\succ_{\text{LEM}}$, 111
$\uparrow$, 14
$\vee$, 14
$\wedge$, 14
b, 119
$d_T V$, 41
e, 12
f-transitivity, 88
h, 8
i, 14
$v^\top$, 15
$v_\perp$, 15
w, 8
#P-complete, 31
Łukasiewicz t-norm, 85

absorbing element, 85
admissible partition, 31
aggregation operators, 84
algebraic product, 85
antichain, 8
antisymmetry, 7
associativity, 84
attribute vector, 8
average rank, 4, 61

balance constant, 119
Bonnet and Pouzet lemma, 16

chain, 8

class cardinalities, 42
classification
    monotone, 27
    problem, 26
    rule, 27
closed walk, 11
commutativity, 84
complete, 8
complete order, 8
complete preorder, 8
conjunctor, 88
constant amortized time, 30
copula, 85
cover, 9
covering relation, 10
criterion, 9
cycle, 11
cycle-transitivity, 92

degree of preference, 83
directed cycle, 11
directed edge, 10
directed graph, 10
directed path, 11
directed walk, 11
directed weighted graph, 11
disjoint sum, 13
distributive lattice, 15
doubt, 28
downset, 14
drastic product, 85
dual lower bound function, 92

edge, 10
ergodic, 41
extension, 12
    linear, 12
    weak order, 42

filter, 14
Frank family, 85
fuzzy preference relation, 88
fuzzy relation, 87

glueing operation, 20
graph, 10
Gray code, 30

Hasse diagram, 10
head, 10

ideal, 14
    lattice, 15
    number, 38
immediate predecessors, 19
immediate successors, 19
incomparable, 3, 7
indifference, 83
interval order, 13
ipsodual relation, 83
isolated element, 10
isomorphic, 16

join, 14
join irreducible, 14
jump number, 30

ladders with broken rungs, 120
lattice, 14
lattice of ideals, 15
    construction, 17
learning example, 27
LEM cycle, 5, 111
LEM relation, 111
linear extension, 12
    average number, 33
    enumeration, 29
    lower bound, 32
    number, 30, 38
    upper bound, 32
linear extension majority relation, *see* LEM relation

linear order relation, 8
linear sum, 13
linearly ordered set, 8

machine learning, 25
Markov chain, 40
    mixing time, 41
    precision, 41
Markov chain Monte Carlo method, 40
max-optimal, 124
maximal chain, 16
maximal element, 9
maximum-length path, 16
MCMC method, 40
meet, 14
meet irreducible, 14
minimal element, 9
minimum cutting level, 122
minimum operator, 85
moderate growth, 85
moderate product transitivity, 96
moderate stochastic transitivity, 96
monotone, 27
monotone classification, 27
monotone classification algorithm, 27
monotone data set, 28
monotonicity, 27
mutual rank probability, 4, 67
mutual rank probability relation, 100

neutral element, 84
node, 12
noise, 28
noise-free data sets, 28

object, 1
order relation, 7
    complete, 8
    linear, 8
    partial, 7
    total, 8
    weak order, 8
order-convex, 14
ordered sorting, 27
ordered sorting algorithm, 27

partial order relation, 7
partial stochastic transitivity, 95
partially ordered set, *see* poset
path, 11
    length, 11
poset, 3, 7
    height, 8
    informal introduction, 3
    number (non-isomorphic), 113
    size, 8
    width, 8
predecessors, 19
probabilistic relation, 83
product transitivity, 96
property, 1

quasi-copula, 85

rank, 61
rank probability, 4, 61
ranking, *see* linear extension
ranking algorithm, 27
rapidly mixing chain, 41
reciprocal relation, 83
reflexivity, 7
reversed preference, 28
root, 12

self-dual, 93
series-parallel poset, 13
sink, 15
source, 15
spanning tree, 12
stationary distribution, 41
Stirlings approximation, 138

strict cut, 120
strong product transitivity, 96
strong stochastic transitivity, 96
successors, 19
supervised learning, 26
synthetic data sets, 28

t-conorm, 85
t-norm, 84
tail, 10
topological sort, *see* linear extension
total order, 8
total preorder, 8
total variation distance, 41
transitive closure, 10
transitive reduction, 11
transitive relation, 86
transitivity, 7
tree, 11
tree of ideals, 18
triangle inequality, 95

unsupervised learning, 26
upper bound function, 91
upset, 14

value, 1
Varol and Rotem algorithm, 29
vertex, 10

walk, 11
    length, 11
weak order extension, 42
weak order relation, 8
weak product transitivity transitivity, 96
weak stochastic transitivity, 96
weight, 2, 11
weight function, 11
worst balanced poset, 119

Bibliography

[1] M. Aigner. *Combinatorial Search*. Wiley-Teubner, Chichester, 1988.

[2] P. Baldy, M. Morvan, and E. Thierry. Extending a partially ordered set: links with the lattice of ideals. *Order*, 16:305–312, 1999.

[3] G. Beliakov. Monotone approximation of aggregation operators using least squares splines. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10:659–676, 2002.

[4] A. Ben-David. Automatic generation of symbolic multi-attribute ordinal knowledge-based DSSs: Methodology and applications. *Decision Sciences*, 23:1357–1372, 1992.

[5] A. Ben-David. Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning*, 9:29–43, 1995.

[6] A. Ben-David, L. Sterling, and Y. Pao. Learning and classification of monotonic ordinal concepts. *Computational Intelligence*, 5:45–49, 1989.

[7] K. Bertet, J. Gustedt, and M. Morvan. Weak-order extensions of an order. *Theoretical Computer Science*, 304:249–268, 2003.

[8] L. Bianco, P. Dell'olmo, and S. Giordani. An optimal algorithm to find the jump number of partially ordered sets. *Computational Optimization and Applications*, 8:197–210, 1997.

[9] G. Birkhoff. *Lattice Theory*. American Mathematical Society Colloquium Publications Vol. 25, Providence, Rhode Island, 1973.

[10] R. Bonnet and M. Pouzet. Extensions et stratifications d'ensembles dispersés. *Comptes Rendus Mathématique Académie des Sciences*, 628-A:1512–1515, 1969.

[11] J. Brans and P. Vincke. A preference ranking organisation method - (The PROMETHEE method for Multiple Criteria Decision-Making). *Management Science*, 31:647–656, 1985.

[12] G. Brightwell. Balanced pairs in partial orders. *Discrete Mathematics*, 201:25–52, 1999.

[13] G. Brightwell, S. Felsner, and W. Trotter. Balancing pairs and the cross product conjecture. *Order*, 12:327–349, 1995.

[14] G. Brightwell, P. Fishburn, and P. Winkler. Interval orders and linear extension cycles. *Ars Combinatoria*, 36:283–288, 1993.

[15] G. Brightwell, H. Prömel, and A. Steger. The average number of linear extensions of a partial order. *Journal of Combinatorial Theory, Series A*, 73:193–206, 1996.

[16] G. Brightwell and P. Winkler. Counting linear extensions is #P-complete. *Order*, 8:225–242, 1992.

[17] G. Brinkmann, J. Degraer, and K. De Loof. Rehabilitation of an unloved child: semi-splaying. *Software - Practice & Experience*, 39:33–45, 2009.

[18] G. Brinkmann and B. McKay. Posets on up to 16 points. *Order*, 19:147–179, 2002.

[19] R. Brüggemann, E. Halfon, and C. Bücherl. Theoretical base of the program "Hasse". *GSF-Bericht 20/95, GSF-Forschungszentrum für Umwelt und Gesundheit, Neuherberg*, 1995.

[20] R. Brüggemann, D. Lerche, and P. Sørensen. First attempts to relate structures of Hasse diagrams with mutual probabilities. Technical Report 479, The 5th workshop held at the National Environmental Research Institute (NERI), Roskilde, Denmark, 2003.

[21] R. Brüggemann, S. Pudenz, K. Voigt, A. Kaune, and K. Kreimes. An algebraic/graphical tool to compare ecosystems with respect to their pollution, IV: Comparative regional analysis by boolean arithmetics. *Chemosphere*, 38:2263–2279, 1999.

[22] R. Brüggemann, U. Simon, and S. Mey. Estimation of averaged ranks by extended local partial order models. *MATCH Communications in Mathematical and in Computer Chemistry*, 54:489–518, 2005.

[23] R. Brüggemann, P. Sørensen, D. Lerche, and L. Carlsen. Estimation of averaged ranks by a local partial order model. *Journal of Chemical Information and Computer Science*, 44:618–625, 2004.

[24] R. Brüggemann and K. Voigt. Basic principles of hasse diagram technique in chemistry. *Combinatorial Chemistry & High Throughput Screening*, 11:756–769, 2008.

[25] R. Bubley and M. Dyer. Faster random generation of linear extensions. *Discrete Mathematics*, 201:81–88, 1999.

[26] E. Canfield and S. Williamson. A loop-free algorithm for generating the linear extensions of a poset. *Order*, 12:1–18, 1995.

[27] K. Cao-Van and B. De Baets. On the definition and representation of a ranking. *Lecture Notes in Computer Science*, 2561:291–299, 2002.

[28] L. Carlsen, D. Lerche, and P. Sørensen. Improving the predicting power of partial order based qsars through linear extensions. *Journal of Chemical Information and Computer Sciences*, 42:806–811, 2002.

[29] F. Chiclana, F. Herrera, and E. Herrera-Viedma. Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations. *Fuzzy Sets and Systems*, 97:33–48, 1998.

[30] H. Daniëls and M. Velikova. Derivation of monotone decision models from noisy data. *IEEE Transactions on Systems, Man and Cybernetics*, 36:705–710, 2006.

[31] M. Dasgupta and R. Deb. Fuzzy choice functions. *Social Choice and Welfare*, 8:171–182, 1991.

[32] B. Davey and H. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, 2003.

[33] H. David. *The method of paired comparisons*. Griffin, London, 1963.

[34] B. De Baets and H. De Meyer. Transitivity frameworks for reciprocal relations: cycle-transitivity versus FG-transitivity. *Fuzzy Sets and Systems*, 152:249–270, 2005.

[35] B. De Baets and H. De Meyer. On the cycle-transitive comparison of artificially coupled random variables. *International Journal of Approximate Reasoning*, 47:306–322, 2008.

[36] B. De Baets, H. De Meyer, and K. De Loof. The omnipresence of cycle-transitivity in the comparison of random variables. In D. Dubois, M. Lubiano, H. Prade, M. Gil, P. Grzegorzewski, and O. Hryniewicz, editors, *Soft Methods for Handling Variability and Imprecision*, volume 48 of *Advances in Soft Computing*, pages 299–305, Berlin, 2008. Springer-Verlag.

[37] B. De Baets, H. De Meyer, and K. De Loof. On the cycle-transitivity of the mutual rank probability relation of a poset. *Fuzzy Sets and Systems*, submitted.

[38] B. De Baets, H. De Meyer, B. De Schuymer, and S. Jenei. Cyclic evaluation of transitivity of reciprocal relations. *Social Choice and Welfare*, 26:217–238, 2006.

[39] K. De Loof, B. De Baets, and H. De Meyer. On the random generation and counting of weak order extensions of a poset with given class cardinalities. *Information Sciences*, 177:220–230, 2007.

[40] K. De Loof, B. De Baets, and H. De Meyer. A hitchhiker's guide to poset ranking. *Combinatorial Chemistry & High Throughput Screening*, 11:734–744, 2008.

[41] K. De Loof, B. De Baets, and H. De Meyer. On the random generation of monotone data sets. *Information Processing Letters*, 107:216–220, 2008.

[42] K. De Loof, B. De Baets, and H. De Meyer. Counting linear extension majority cycles in posets on up to 13 points. *Computers & Mathematics with Applications*, submitted.

[43] K. De Loof, H. De Meyer, and B. De Baets. Exploiting the lattice of ideals representation of a poset. *Fundamenta Informaticae*, 71:309–321, 2006.

[44] K. De Loof, H. De Meyer, and B. De Baets. Graded stochastic dominance as a tool for ranking the elements of a poset. In J. Lawry, E. Miranda, A. Bugarin, S. Li, M. Gil, P. Grzegorzewski, and O. Hyrniewicz, editors, *Soft Methods for Integrated Uncertainty Modelling*, Advances in Soft Computing, pages 53–60, Berlin, 2006. Springer-Verlag.

[45] H. De Meyer, B. De Baets, and B. De Schuymer. On the transitivity of comonotonic and countermonotonic comparison of random variables. *Journal of Multivariate Analysis*, 98:177–193, 2007.

[46] B. De Schuymer, H. De Meyer, and B. De Baets. Cycle-transitive comparison of independent random variables. *Journal of Multivariate Analysis*, 96:352–373, 2005.

[47] B. De Schuymer, H. De Meyer, and B. De Baets. Optimal strategies for equal-sum dice games. *Discrete Applied Mathematics*, 154:2565–2576, 2006.

[48] B. De Schuymer, H. De Meyer, and B. De Baets. Extreme copulas and the comparison of ordered lists. *Theory and Decision*, 62:195–217, 2007.

[49] B. De Schuymer, H. De Meyer, B. De Baets, and S. Jenei. On the cycle-transitivity of the dice model. *Theory and Decision*, 54:261–285, 2003.

[50] J. Doignon, B. Monjardet, M. Roubens, and P. Vincke. Biorder families, valued relations and preference modelling. *Journal of Mathematical Psychology*, 30:435–480, 1986.

[51] B. Dutta and J. Laslier. Comparison functions and choice correspondences. *Social Choice and Welfare*, 16:513–532, 1999.

[52] M. Dyer, R. Frieze, and A. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the Association for Computing Machinery*, 38:1–17, 1991.

[53] M. Dyer, L. Goldberg, C. Greenhill, and M. Jerrum. On the relative complexity of approximate counting problems. *Algorithmica*, 38:471–500, 2003.

[54] K. Ewacha, P. Fishburn, and W. Gehrlein. Linear extension majority cycles in height-1 orders. *Order*, 6:313–318, 1990.

[55] K. Ewacha, I. Rival, and N. Zaguia. Approximating the number of linear extensions. *Theoretical Computer Science*, 175:271–282, 1997.

[56] P. Fishburn. Intransitive indifference in preference theory: a survey. *Operations Research*, 18:207–228, 1970.

[57] P. Fishburn. Binary choice probabilities: on the varieties of stochastic transitivity. *Journal of Mathematical Psychology*, 10:327–352, 1973.

[58] P. Fishburn. On the family of linear extensions of a partial order. *Journal of Combinatorial Theory, Series B*, 17:240–243, 1974.

[59] P. Fishburn. On linear extension majority graphs of partial orders. *Journal of Combinatorial Theory, Series B*, 21:65–70, 1976.

[60] P. Fishburn. Proportional transitivity in linear extensions of ordered sets. *Journal of Combinatorial Theory, Series B*, 41:48–60, 1986.

[61] P. Fishburn and W. Gehrlein. A comparative analysis of methods for constructing weak orders from partial orders. *Journal of Mathematical Sociology*, 4:93–102, 1975.

[62] B. Ganter, G. Hafner, and W. Poguntke. On linear extensions of ordered sets with a symmetry. *Discrete Mathematics*, 63:153–156, 1987.

[63] B. Ganter, R. Wille, and C. Franzke. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin, 1991.

[64] J. García-Lapresta and B. Llamazares. Aggregation of fuzzy preferences: some rules of the mean. *Social Choice and Welfare*, 17:673–690, 2000.

[65] J. García-Lapresta and B. Llamazares. Majority decisions based on difference of votes. *Journal of Mathematical Economics*, 35:463–481, 2001.

[66] W. Gehrlein. Frequency estimates for linear extension majority cycles on partial orders. *RAIRO-Operations Research*, 25:359–364, 1991.

[67] W. Gehrlein. The effectiveness of weighted scoring rules when pairwise majority rule cycles exist. *Mathematical Social Sciences*, 47:69–85, 2004.

[68] W. Gehrlein and P. Fishburn. Linear extension majority cycles for partial orders. *Annals of Operations Research*, 23:311–322, 1990.

[69] W. Gehrlein and P. Fishburn. Linear extension majority cycles for small ($n \leq 9$) partial orders. *Computers & Mathematics with Applications*, 20:41–44, 1990.

[70] M. Habib, R. Medina, L. Nourine, and G. Steiner. Efficient algorithms on distributive lattices. *Discrete Applied Mathematics*, 110:169–187, 2001.

[71] M. Habib and L. Nourine. Tree structure for distributive lattices and its applications. *Theoretical Computer Science*, 165:391–405, 1996.

[72] E. Herrera-Viedma, F. Herrera, F. Chichlana, and M. Luque. Some issues on consistency of fuzzy preference relations. *European Journal of Operational Research*, 154:98–109, 2004.

[73] M. Huber. Fast perfect sampling from linear extensions. *Discrete Mathematics*, 306:420–428, 2006.

[74] J. Kacprzyk, M. Fedrizzi, and H. Nurmi. Group decision making and consensus under fuzzy preferences and fuzzy majority. *Fuzzy Sets and Systems*, 49:21–31, 1992.

[75] J. Kacprzyk, H. Nurmi, and M. Fedrizzi. *Consensus under Fuzziness*. Kluwer Academic Publishers, Boston, 1996.

[76] J. Kahn and Y. Yu. Log-concave functions and poset probabilities. *Combinatorica*, 18:85–99, 1998.

[77] A. Kalvin and Y. Varol. On the generation of all topological sortings. *Journal of Algorithms*, 4:150–162, 1983.

[78] S. Kislitsyn. Finite partially ordered sets and their associated sets of permutations. *Matematicheskiye Zametki*, 4:511–518, 1968.

[79] E. Klement, R. Mesiar, and E. Pap. *Triangular Norms*, volume 8 of *Trends in Logic, Studia Logica Library*. Kluwer Academic Publishers, Dordrecht, 2000.

[80] D. Knuth and J. Szwarcfiter. A structured program to generate all topological sorting arrangements. *Information Processing Letters*, 2:153–157, 1974.

[81] Y. Koda and F. Ruskey. A Gray code for the ideals of a forest poset. *Journal of Algorithms*, 15:324–340, 1993.

[82] S. Lahiri. Axiomatic characterizations of threshold choice functions for comparison functions. *Fuzzy Sets and Systems*, 132:77–82, 2002.

[83] D. Lerche and P. Sørensen. Evaluation of the ranking probabilities for partial orders based on random linear extensions. *Chemosphere*, 53:981–992, 2003.

[84] D. Lerche, P. Sørensen, and R. Brüggemann. Improved estimation of the ranking probabilities in partial orders using random linear extensions by approximation of the mutual ranking probability. *Journal of Chemical Information and Computer Science*, 43:1471–1480, 2003.

[85] S. Lievens. Studie en implementatie van instantiegebaseerde algoritmen voor gesuperviseerd rangschikken. Master's thesis, Ghent University, 2004.

[86] S. Lievens, B. De Baets, and K. Cao-Van. A probabilistic framework for the design of instance-based supervised ranking algorithms in an ordinal setting, 2008.

[87] S. Lievens and B. De Baets. Supervised ranking in the WEKA environment. *Information Sciences*, in press.

[88] J. Marichal, P. Meyer, and M. Roubens. Sorting multi-attribute alternatives: The TOMASO method. *Computers & OR*, 32:861–877, 2005.

[89] P. Matthews. Generating a random linear extension of a partial order. *The Annals of Probability*, 19:1367–1392, 1991.

[90] C. McCartin. An improved algorithm for the jump number problem. *Information Processing Letters*, 79:87–92, 2001.

[91] B. Mirkin. *Mathematical Classification and Clustering*. Kluwer Academic Publishers, Dordrecht, 1996.

[92] R. Nelsen. *An Introduction to Copulas*, volume 139 of *Lecture Notes in Statistics*. Springer-Verlag, New York, second edition edition, 2005.

[93] N.N. Signale aus der natur, 10 Jahre ökologisches Wirkungskataster Baden-Württemberg. pages 1–63. Ettlingen: Kraft Druck GmbH, 1994.

[94] H. Nurmi. *Comparing Voting Systems*. Reidel, Dordrecht, 1987.

[95] A. Ono and S. Nakano. Constant time generation of linear extensions. *Lecture Notes in Computer Science*, 3623:445–453, 2005.

[96] G. Patil and C. Taillie. Multiple indicators, partially ordered sets, and linear extensions: Multi-criterion ranking and prioritization. *Environmental and Ecological Statistics*, 11:199–228, 2004.

[97] M. Peczarski. New results in minimum-comparison sorting. *Algorithmica*, 40:133–145, 2004.

[98] M. Peczarski. *Computer assisted research of posets*. PhD thesis, University of Warsaw, 2006.

[99] R. Potharst and M. van Wezel. Generating artificial data with monotonicity constraints. Technical report, Erasmus University Rotterdam, 2005.

[100] G. Pruesse. Generating the linear extensions of certain posets by adjacent transpositions. *SIAM Journal on Discrete Mathematics*, 4:413–422, 1991.

[101] G. Pruesse and F. Ruskey. Gray codes from antimatroids. *Order*, 10:239–252, 1993.

[102] G. Pruesse and F. Ruskey. Generating linear extensions fast. *SIAM Journal on Computing*, 23:373–386, 1994.

[103] M. Rademaker, B. De Baets, and H. De Meyer. Loss optimal monotone relabeling of noisy multi-criteria data sets. *Information Sciences*, 179:4089–4096, 2009.

[104] M. Rademaker, B. De Baets, and H. De Meyer. Optimal monotone relabeling of partially non-monotone ordinal data. *European Journal of Operations Research*, submitted.

[105] F. Ruskey. Research problem 90. *Discrete Mathematics*, 70:111–112, 1988.

[106] F. Ruskey. Generating linear extensions of posets by transpositions. *Journal of Combinatorial Theory, Series A*, 54:77–101, 1992.

[107] F. Ruskey and C. Savage. Hamilton cycles that extend transposition matchings in Cayley-graphs of Sn. *SIAM Journal on Discrete Mathematics*, 6:152–166, 1993.

[108] C. Schneeweiss. *Planung 1, Systemanalytische und entscheidungstheoretische Grundlagen*. Springer-Verlag, Berlin, 1991.

[109] U. Simon, R. Brüggeman, and S. Pudenz. Aspects of decision support in water management - example Berlin and Potsdam (Germany) I - spatially differentiated evaluation. *Water Research*, 38:1809–1816, 2004.

[110] U. Simon, R. Brüggeman, and S. Pudenz. Aspects of decision support in water management - example Berlin and Potsdam (Germany) II - improvement of management strategies. *Water Research*, 38:4085–4092, 2004.

[111] M. Squire. Enumerating the ideals of a poset. *North Carolina State University*, 1995.

[112] G. Stachowiak. Hamilton paths in graphs of linear extensions for unions of posets. *SIAM Journal on Discrete Mathematics*, 5:199–206, 1992.

[113] S. Switalski. Rationality of fuzzy reciprocal preference relations. *Fuzzy Sets and Systems*, 107:187–190, 1999.

[114] M. Sysło. Minimizing the jump-number for partially ordered sets: A graph-theoretic approach. *Order*, 1:7–19, 1984.

[115] T. Tanino. Non-conventional preference relations in decision making. In *Lecture Notes in Economics and Mathematical Systems*, volume 301, chapter Fuzzy preference relations in group decision making, pages 54–71. Springer-Verlag, Berlin, 1988.

[116] Y. Varol and D. Rotem. An algorithm to generate all topological sorting arrangements. *The Computer Journal*, 24:83–84, 1981.

[117] A. Verkeyn, D. Botteldooren, and B. De Baets. Genetic learning of fuzzy integrals accumulating human-reported environmental stress. *Applied Soft Computing*, to appear.

[118] W. Waegeman and B. De Baets. A transitivity analysis of bipartite rankings in pairwise multi-class classification. *Information Sciences*, submitted.

[119] D. West. Generating linear extensions by adjacent transpositions. *Journal of Combinatorial Theory, Series B*, 57:58–64, 1993.

[120] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, Elsevier, San Fransisco, 2005.

[121] Y. Yu. On proportional transitivity of ordered sets. *Order*, 15:87–95, 1998.