



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services Branch

Direction des acquisitions et
des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

395 Wellington Street

395, rue Wellington

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Linear Extensions of Ordered Sets

by

Kevin Ewacha

A Ph. D. Thesis

submitted to the School of Graduate Studies and Research

in partial fulfillment of the requirements for

the degree of Doctor of Philosophy in Mathematics*

University of Ottawa

Ottawa, Ontario

Canada

May 12, 1993

*The Ph. D. Program is a joint program with
Carleton University, administered by the Ottawa-Carleton

Institute of Mathematics and Statistics

© 1993 by Kevin Ewacha



National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Vous le *Votre référence*

Cela le *Notre référence*

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-89593-4

Canada



UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Acknowledgements

I would very much like to thank my supervisor Ivan Rival, for all his guidance and support over the years. I never would have finished if it was not for his constant reassurances that I could do so, when I was sure that I could not. He was correct. Thank you.

I would also like to thank Nejib Zaguia for his support and guidance, both while Dr. Rival was away on sabbatical, and after his return.

Finally, I would like to thank Alexander Sidorenko for pointing me in the right direction when the "approximations" were at a standstill.

Contents

1	Introduction	1
1.1	Definitions and Notation — “Minimals”	1
1.2	Further Results	11
1.3	Linear Extensions	19
1.3.1	Finding Linear Extensions	19
1.3.2	Linear Extension Count	21
2	Main Results	27
2.1	Linear Extensions	27
2.1.1	Approximation	27
2.1.2	Enumeration	29
2.2	Majority Cycles	32
2.3	Scheduling	32
2.4	Genus and Diagram Invariance	33
2.5	Cataloguing Orders	33
3	Approximating Linear Extension Count	35
3.1	Upper Bound	36
3.2	Lower Bound	39

3.3	Alternating Bounds	49
3.4	Critical Orders and Coefficients	59
4	Linear Extensions of Width Two Orders	67
4.1	Graph Construction	68
4.2	Counting Linear Extensions	75
4.3	Width k Orders	84
5	Unimodality and Probability	89
5.1	Unimodal Sequences	90
5.2	Calculating Probabilities	94
5.3	Extremal Width Two Orders	98
6	Linear Extension Majority Cycles	103
6.1	Background	104
6.2	Linear Extensions of Height two Orders	106
6.3	Miscellany	109
7	Scheduling	113
7.1	Introduction	113
7.2	Main Results	118
7.3	General Flow Shop Problems	126
7.4	Concluding Remarks	128
8	Genus and Diagram Invariance	131
8.1	Planarity and Genus	131
8.2	Genus Zero Orders	133
8.3	Higher Genus	138

<i>CONTENTS</i>	iii
9 Cataloguing Ordered Sets	145
9.1 Generating Lists of Orders	147
9.2 Orders Without Irreducibles	152
Bibliography	157

List of Figures

1.1	Diagrams	3
1.2	Diagram, covering graph and comparability graph	4
1.3	Series-parallel order	11
1.4	Several diagrams of the cube	12
1.5	Alternate orientations of N	13
1.6	Lexicographic decomposition	15
1.7	An N -free order which is not series parallel	16
1.8	Extremal width two orders for $\delta(P) = \frac{1}{3}$	18
1.9	Greedy and level linear extensions.	20
3.1	Three-element critical orders	38
3.2	Four-element critical orders	50
3.3	Approximation example	52
3.4	Five-element critical orders	65
4.1	A width two order and its graph	70
4.2	Linear extension calculation	82
4.3	Extension calculation for a decomposable order	84
4.4	Extension calculation for width three order	88

5.1	Unimodal sequences	91
5.2	Extremal width two orders for $\delta(P) = \frac{1}{3}$	99
5.3	The cases for the proof of Theorem 5.3	101
6.1	Height two order with majority cycle $1 >' 2 >' 3 >' 1$	104
6.2	Nine-element orders with majority cycles	104
6.3	"Natural" linear extension of N	105
6.4	Linear extension calculation for height two orders	107
8.1	A nonplanar order with a planar covering graph	134
8.2	The cube on the sphere without edge crossings	134
8.3	Double cube	135
8.4	Embedding of double cube on a genus zero surface	137
8.5	Lifting of the double cube	138
8.6	Polygon model of surface for genus one order	141
9.1	Orders without irreducible elements	146
9.2	Several possible labellings of an unlabelled order	148

Chapter 1

Introduction

“Ordered Sets”, as a subject, has only been studied for a short time, as far as Mathematics goes. Indeed, basic results of great importance to the subject were only proved quite recently; for example, [Szpilrajn 30] and [Dilworth 50]. Nonetheless, the subject is very important, with a wide variety of applications such as sorting and scheduling.

1.1 Definitions and Notation — “Minimals”

A (*partially*) *ordered set*, or simply *order*, $P = (X, <)$, consists of a set X and a binary relation $<$, *less than*, defined on $X \times X$, which is “transitive” and “asymmetric”. We follow the common practice of referring to P itself as the set of elements, rather than X : instead of writing (correctly) $a \in X$, we write $a \in P$. The other relational symbols, such as $\neq, >, \neq, \leq, \geq$, etc. are used with obvious meanings: $a > b$ if and only if $b < a$; $a \leq b$ if and only if $a < b$ or $a = b$. Then, for $a, b, c \in P$:

- $<$ is *transitive*: $a < b$ and $b < c$ imply $a < c$;

- $<$ is asymmetric: $a < b$ implies $b \not< a$.

Example 1.1

(i) $P = (\{a, b, c, d\}, \{a < c, b < c, b < d\})$, for reasons which will soon be apparent, this order is usually referred to as \mathbb{N} .

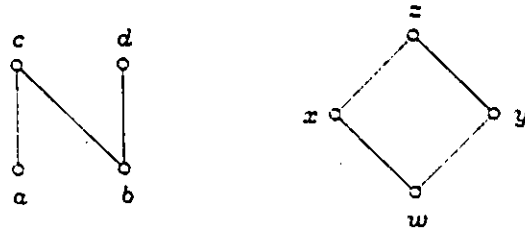
(ii) $P = (\{w, x, y, z\}, \{w < x, w < y, w < z, x < z, y < z\})$.

The symbol " $<$ ", used to denote the relation in the order is the same symbol that is used for comparing real numbers; this rarely results in any confusion as the meaning is usually clear from the context. However, if necessary, " $<$ " may be indexed by the order. For example, suppose that an order P is defined on the set $\{1, 2, \dots, n\}$. Then, $1 <_P 2$ compares two elements of the order, while $1 <_{\mathbb{R}} 2$ compares the real numbers 1 and 2 (as elements of the order \mathbb{R}). Indexing may also be necessary when considering several different orders on the same set of elements.

While an order may be infinite, our interest is finite orders. Accordingly, unless otherwise stated, all orders are assumed to be finite.

Elements a, b in P are called *comparable* if $a < b$ or $b < a$; otherwise, they are *incomparable*, denoted by $a \parallel b$. Say that a is *covered* by b , written $a < b$, if $a < b$ and there does not exist $c \in P$ with $a < c < b$. That is, there is nothing "between" a and b . Equivalently, b *covers* a , denoted $b > a$. Call b an *upper cover* of a , and a a *lower cover* of b . The *dual* of an order P , is the order P' , on the same set of elements, but with all comparabilities reversed. That is $a < b$ in P' if and only if $a > b$ in P .

An order may be specified by a list of its comparabilities or covering relations; however, we usually present orders pictorially, using a graphical



Diagrams of the orders of Example 1.1.

Figure 1.1: Diagrams

representation called the “diagram”. (See, for example, Figure 1.1.) The *covering graph* of an order P , is the (undirected) graph whose vertices are the elements of the order, with the vertices corresponding to $a, b \in P$ adjacent precisely if $a < b$ or $a > b$. The *diagram* of P is a drawing of its covering graph such that if $a < b$ then a is placed lower than b (a has a smaller y -coordinate in the plane than b), with the edge joining a and b rising monotonically from a to b . The diagram is often referred to as the order itself.

In addition, we may consider the *comparability graph* of an order, in which a pair of elements are adjacent simply if they are comparable. See Figure 1.2, for an example of an order (its diagram), with its covering and comparability graphs. In general, the comparability graph is more difficult to read than the diagram (or covering graph) as it contains a much larger number of edges.

Call $a \in P$ *maximal* if there does not exist $b \in P$ with $a < b$; a has no upper covers. Similarly, $b \in P$ is *minimal* if it has no lower covers. The sets of minimal and maximal elements of P are denoted by $\min(P)$ and

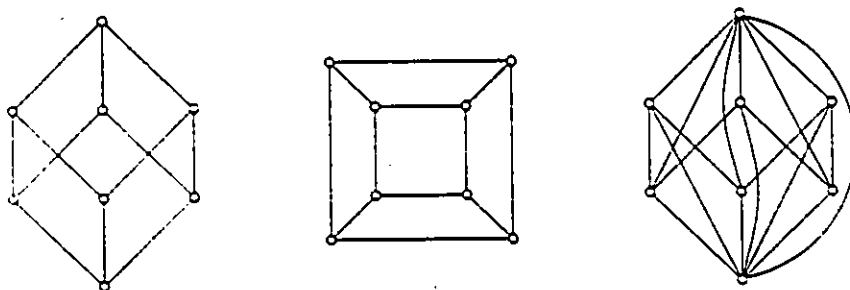


Figure 1.2: Diagram, covering graph and comparability graph

$\max(P)$, respectively. If P has a unique maximal element, it is called the *top* element of P . A unique minimal element is called the *bottom* element of P . Both the top and bottom elements (if they exist) are examples of *splitting elements*, elements which are comparable to all other elements. In contrast, an *isolated element* of an order is one which is comparable to no other element of the order. The term “isolated” results from the fact that an isolated element is an isolated vertex in the covering and comparability graphs.

Example 1.2 In the order of Example 1.1 (ii), all pairs of elements are comparable, except for x and y ; all comparabilities are covering relations except that $w \not\prec z$; and w, z are the top and bottom elements of P , respectively.

The order $Q (Q = (Y, <))$ is a *suborder* of the order $P (P = (X, <))$, if $Q \subseteq P (Y \subseteq X)$ and $<$ on Q is the restriction of $<$ on P to the elements of Q . Given a subset Q of P , the *induced suborder* on Q is the suborder of P on the set Q . For a subset $A \subset P$, let $P \setminus A$ denote the induced suborder

on the set $P \setminus A$. If $x \in P$ then $P \setminus \{x\}$ (or simply $P \setminus x$) is the order produced by removing the element x and all comparabilities involving x .

An order is a *linear order* (is *linearly ordered*), also called a *chain*, if all pairs of elements are comparable. A *chain* is also a linearly ordered suborder of an order. An *antichain* is an order all of whose elements are pairwise incomparable, as well as such a suborder of an order. The n -element chain is denoted by n ; the n -element antichain by n . A chain C , in an order P , is a *maximal* or *saturated chain* if there does not exist a larger chain in P , containing C . That is, if $a \in P \setminus C$ then a is incomparable with some element of C : adding a to C yields a suborder of P which is not a chain. A maximal chain is not to be confused with a *maximum (sized) chain* in P , which is a chain with the greatest possible number of elements. Clearly, a maximum sized chain is necessarily maximal; however, a maximal chain need not have maximum size among all chains. *Maximal* and *maximum antichains* are similarly defined. The set of maximal elements of an order is an antichain, as is the set of minimal elements: both are maximal antichains.

The *height* of P is defined to be the cardinality of a maximum sized chain in P . (Some authors define the height to be one less than the cardinality of a maximum sized chain.) The *width* of P is the cardinality of a maximum sized antichain in P . For an element x in P , define the *height* of x by

$$\text{height}(x) = \max\{|C| : C \text{ a chain in } P \text{ with top element } x\},$$

and define the *depth* of x by

$$\text{depth}(x) = \max\{|C| : C \text{ a chain in } P \text{ with bottom element } x\}.$$

In general, there is no simple relationship between the height and depth of an element; however, $height(x)$ in P equals $depth(x)$ in the dual of P .

Theorem 1.1 [Dilworth 50] *In a finite order P the minimum number of disjoint chains whose set union is P equals the width of P .*

A set of disjoint chains whose (set) union is the order P is called a *chain decomposition* of P . In general, a minimum chain decomposition is not unique. If the chains are not disjoint then P is simply *covered* by the set of chains. Allowing the chains to overlap cannot reduce the number of chains required to cover P since no chain may contain more than one element from any (maximal) antichain.

A (*partial*) *extension* of an order is an order on the same set of elements which includes all the same comparabilities, plus, possibly, some extra. That is, Q is an extension of P , if P and Q are defined on the same set of elements, and $a < b$ in P implies $a < b$ in Q . Let $a, b \in P$ be incomparable. Then $P(a < b)$ is the extension of P formed by adding the relation $a < b$ to P and taking the *transitive closure*; that is, adding all other relations required by transitivity. If an extension of the order P is a linear order then it is called a *linear extension* of P .

Example 1.3 The order N (Example 1.1 (ii), and Figure 1.1) has five linear extensions, they are (listing covering relations only)

1. $a < b < c < d$,
2. $a < b < d < c$,
3. $b < a < c < d$,
4. $b < a < d < c$,
5. $b < d < a < c$.

Theorem 1.2 [Szpilrajn 30] *Every order has a linear extension.*

The set of all linear extensions of P is denoted by $L(P)$; their number by $e(P)$. A very important problem in the theory of ordered sets is computing the number of linear extensions of an order. (Of course, it is trivial that a finite order has a finite number of linear extensions.) From a theoretical point of view, the problem is trivial: the set of all linear extensions of an order is easily generated, e.g. by “topological sorting”, and counted. However, from a computational point of view, calculating the number of linear extensions is very difficult.

Theorem 1.3 [Brightwell/Winkler 91] *Computing the number of linear extensions of an order is #P-complete.*

The class #P (read “number-P”), defined in [Valiant 79], is the class of all problems solvable by a nondeterministic polynomial time Turing machine, which also outputs the number of accepting states. A problem in #P is called #P-complete, if the existence of a polynomial time (deterministic) solution for that problem implies the existence of such a solution for all problems in #P. Problems which are #P-complete are generally more difficult than NP-complete problems. The reason for this is that #P-complete problems typically count the number of solutions of an NP-complete problem, such as satisfiability. Counting linear extensions is a rare example of a #P-complete problem where the decision problem — is a given ordering of the elements a linear extension of the order? — is in the class P.

There are a very small number of special classes of orders for which the number of linear extensions may be computed directly, or by a recursive

formula.

Example 1.4 Some special cases for computing linear extensions:

- the n -element chain: $e(n) = 1$;
- the n -element antichain: $e(n) = n!$;
- $P = P_1 + P_2$ is a "disjoint sum" (see page 10) of P_1 and P_2 :

$$e(P) = \binom{|P|}{|P_1|} e(P_1)e(P_2);$$

- $P = P_1 * P_2$ is a "linear sum" (see page 10) of P_1 and P_2 :

$$e(P) = e(P_1)e(P_2);$$

- $Z_n = (\{a_1, a_2, \dots, a_n\}, \{a_{2i-1} < a_{2i}, a_{2i} > a_{2i+1}\})$ the n -element zig-zag:

$$e(Z_n) = \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n-1}{2i} e(Z_{2i})e(Z_{n-1-2i});$$

- $C_{2n} = (\{a_1, a_2, \dots, a_{2n}\}, \{a_{2i-1} < a_{2i}, a_{2i} > a_{2i+1}, a_{2n} > a_1\})$ the $2n$ -element cycle or crown: $e(C_{2n}) = n e(Z_{2n-1})$.

If a and b are incomparable in P then there are linear extensions of P in which $a < b$ and others in which $b < a$. To compare the numbers of such extensions we define

$$\text{prob}(P; a < b) = \frac{e(P(a < b))}{e(P)}.$$

This is the *probability* that a occurs below b in an randomly chosen linear extension of P . For brevity, we also write $\text{prob}(a < b)$.

Related to this is the *preference relation* $<'$, defined by

$$x <' y \quad \text{if} \quad \text{prob}(x < y) > \frac{1}{2}.$$

The relation $<'$ is defined by what happens in the majority of extensions of P . If $\text{prob}(x < y) = 1/2$ then $x \parallel' y$. Clearly, $x < y$ implies $x <' y$ since, in this case, $\text{prob}(x < y) = 1$. If the relation $<'$ defines a valid ordering; that is, one which is transitive, then the resulting order is a partial extension of P . If $<'$ is not transitive, then there exists a sequence of elements $x_1 <' x_2 <' \dots <' x_k <' x_1$. If such a sequence exists, it is called a *linear extension majority cycle* or simply a *majority cycle*. In fact, if an order contains a majority cycle then it contains one with only three elements, the minimum possible.

Theorem 1.4 [Fishburn 76] *There exist orders which contain linear extension majority cycles.*

The idea of probability is generalized. If S is any property which may hold in a linear extension, define $e(P; S)$ to be the number of linear extensions of P in which S holds. Then

$$\text{prob}(P; S) = \frac{e(P; S)}{e(P)}$$

is the probability that S holds in a given linear extension of P .

Theorem 1.5 [Stanley 81] *For $x \in P$, the sequence*

$$(e(P; \text{height}(x) = i))_{i=1}^n$$

is a unimodal sequence. (height(x) is the height of x in the linear extension.)

A sequence is *unimodal* if it consists of an increasing (nondecreasing) segment followed by a decreasing (nonincreasing) segment, either of which may be empty. That is, $(s_i)_{i=1}^n$ is unimodal if there exists j such that

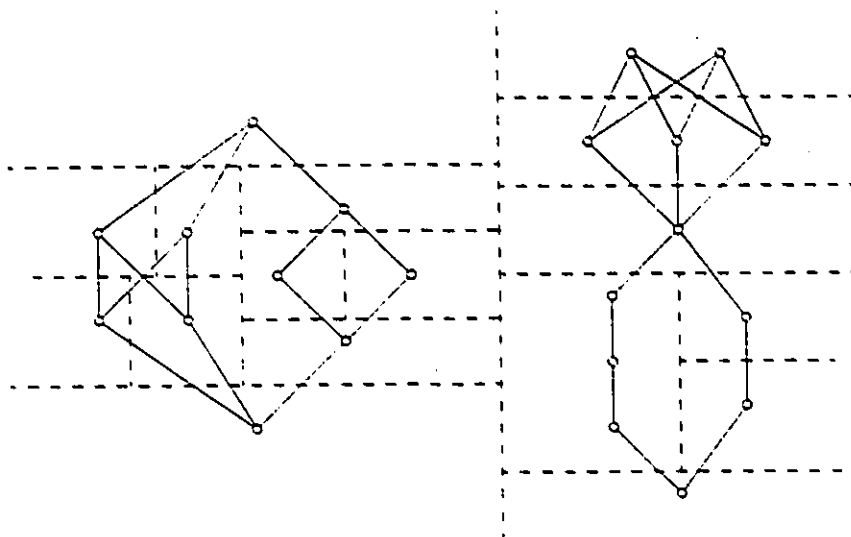
$$s_1 \leq s_2 \leq \cdots \leq s_j \geq s_{j+1} \geq \cdots \geq s_n.$$

Given two orders P and Q , a function $f: P \rightarrow Q$ maps the elements of P to the elements of Q . The function is *isotone* if it is *order preserving*; that is, $f(a) \leq f(b)$ in Q whenever $a < b$ in P . An isotone map is an *isomorphism* if it is one to one and onto, and its inverse is also an isotone map. An *automorphism* is an isomorphism from an order to itself. A *rigid* order is one whose only automorphism is the identity map. If two orders are isomorphic, they are, in effect, the same order, possibly with a relabelling of the elements.

A *series-parallel* order is one which may be constructed, from singletons, by means of the operations “disjoint sum” $+$ and “linear sum” $*$. (See for example, Figure 1.3.) The order $P = P_1 \cup P_2$ is the *disjoint sum* of P_1 and P_2 if the comparabilities in P are precisely those “inherited” from P_1 and P_2 . That is, $a < b$ in P if and only if $a < b$ in P_1 or $a < b$ in P_2 . The *linear sum* $P_1 * P_2$ places P_2 “above” P_1 . It is the partial extension of $P_1 + P_2$ in which all comparabilities of the form $a < b$ (for $a \in P_1$ and $b \in P_2$), and no others, have been added.

Theorem 1.6 [Foulis 70] *A finite ordered set is series-parallel if and only if it contains no subset isomorphic to N .*

An order P is a *linear sum* or is (*linearly*) *decomposable* if there exist orders P_1 and P_2 such that $P = P_1 * P_2$. Otherwise, P is (*linearly*)



$$[1 * [((1+1) * (1+1)) + (1 * (1+1) * 1)] * 1] + [1 * (1 * 1 * 1 + 1 * 1) * 1 * (1+1+1) * (1+1)]$$

Figure 1.3: Series-parallel order

indecomposable. A linear decomposition of P is a linear sum of the form $P = P_1 * P_2 * \dots * P_k$, where P_i is indecomposable. Call the P_i the *indecomposable components* of P .

1.2 Further Results

The diagram considers only covering relations to avoid *nonessential edges*, edges specifying comparabilities deducible by transitivity. If $a < b$ but $a \not\prec b$, then there is a sequence of elements $(c_i)_{i=1}^k$ such that $a \prec c_1 \prec c_2 \prec \dots \prec c_k \prec b$: there is a monotonic path rising from a to b . However, the existence of such a monotonic path implies, by transitivity, that $a < b$.

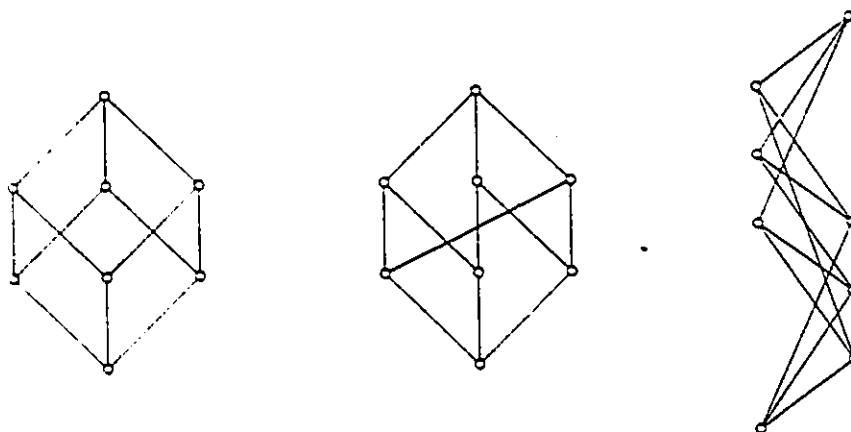
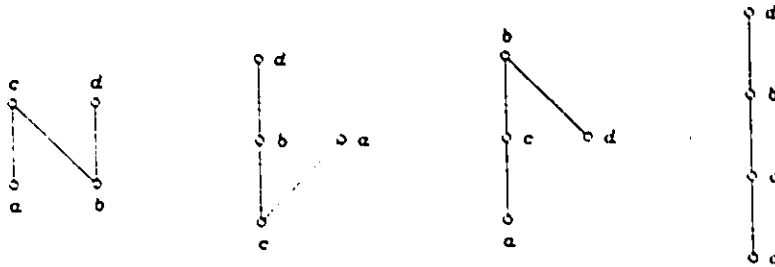


Figure 1.4: Several diagrams of the cube

Therefore, an edge from a to b would be redundant. Such edges are not drawn. Instead, comparabilities are read by following monotonic paths. Note that if b is “above” a , but there is no monotonic path joining them, then $a \not\leq b$. In fact, a is incomparable to b , as $b < a$ is clearly not possible.

The diagram is simply a directed version of the covering graph — the directions decided by the directions of the covering relations — in which the directions are indicated, not by arrows on the edges, but by the relative vertical placement of the vertices. (With the added restriction that the edges are monotonic.) The diagram is not unique since there is some freedom in the placement of the vertices, and in drawing the edges. Figure 1.4 shows several possible diagrams of the cube. However, different diagrams of the same order all “yield” the same directed (covering) graph. In fact, if the diagrams of any two orders yield the same directed graph, the orders are isomorphic: essentially, they are the same order.



All possible orientations, up to duality, of the covering graph of N

Figure 1.5: Alternate orientations of N

An order is called a *planar order* if its covering graph can be drawn without *edge crossings*, points other than vertices where edges intersect. A planar order necessarily has a planar covering graph, the converse is not true. The cube (cf. Figure 1.2) is non planar, yet has a planar covering graph.

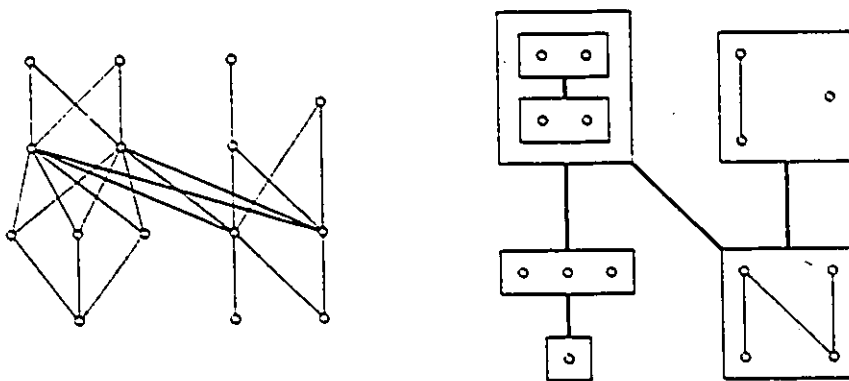
The diagram gives an *orientation* of the covering graph: a directing of the edges corresponding to an order. Not all graphs are orientable; that is, not all graphs are covering graphs. For example, a triangle, a complete graph on three vertices, is not orientable: one edge must be redundant. Frequently, a graph which is orientable has more than one orientation. There are different (nonisomorphic) orders which have the same covering graph. (See Figure 1.5.) In general, the different orientations of a covering graph yield very different orders, having very different properties. It is more interesting to consider which properties such orders may have in common. Call an order theoretic property a *diagram invariant* if, whenever it holds for an order, it holds for all other orders having the same covering graph.

We show that the “genus” of an ordered set is a diagram invariant. This is, in fact, the first and only known nontrivial diagram invariant. (The number of covering relations is an example of a trivial — hence uninteresting — diagram invariant.)

While the “extra” edges in the comparability graph make it more difficult to read (than the diagram), these edges do serve to more precisely define the order. In general the comparability graph of an order has very “few” orientations, compared to the covering graph.

Theorem 1.7 [Kelly 86] *If P and Q have the same comparability graphs then Q can be obtained from P by a sequence of operations of dualizing autonomous subsets of P .*

Linear decomposition is a special case of “lexicographic decomposition”. A subset A of P is *autonomous* if each element of $P \setminus A$ bears the same relation to all elements of A . That is, if $a, b \in A$, $x \in P \setminus A$, then $a < x$ if and only if $b < x$, and $a > x$ if and only if $b > x$ (so $a \parallel x$ if and only if $b \parallel x$). Note however, that $a < x$ says nothing at all about the relation between a and other elements of $P \setminus A$. (In Example 1.1 (ii), the set $\{x, y\}$ is an autonomous set.) An order is *lexicographically decomposable* if it contains a nontrivial autonomous set, one which is not a singleton or the entire order; otherwise the order is *lexicographically indecomposable* or *prime*. The diagrams of lexicographically decomposable orders may be simplified by grouping together the elements of the autonomous sets, and drawing drawing single arcs to represent comparabilities applying to all elements in the autonomous set. Call this simplified diagram a *block diagram* for the order. See Figure 1.6 for an example of a block diagram.



A lexicographically decomposable order and its block diagram.

Figure 1.6: Lexicographic decomposition

Theorem 1.8 [Möhring 84] *Almost all orders are prime, that is,*

$$\lim_{n \rightarrow \infty} \frac{\text{prime } n\text{-element orders}}{n\text{-element orders}} = 1.$$

The *incomparability graph* of an order P , is the (graph) complement of the comparability graph. This is simply the graph on the elements of P in which two vertices are adjacent precisely if their corresponding elements are incomparable in P . If there exists an order Q whose comparability graph is isomorphic to the incomparability graph of P , then call the orders P and Q *complementary*. In the complementary orders P and Q , a pair of elements in P are comparable if and only if the corresponding elements of Q are incomparable. The order Q is called a *complement* of P , denoted \bar{P} . (In fact the complement need not be unique since the covering graph may have more than one orientation.)

A subset B of P is called a *cutset* if it intersects every maximal chain

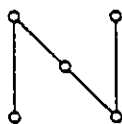


Figure 1.7: An N-free order which is not series parallel

in P . An analogous idea is that of a *fiber*, a subset which intersects every maximal antichain in the order.

Theorem 1.9 [Grillet 69] *In a finite order P , every maximal chain intersects every maximal antichain if and only if P is N-free.*

That is, every maximal antichain is a cutset and every maximal chain is a fiber if and only if P is “N-free”.

Say that P is N-free if P contains no cover-preserving subset isomorphic to N. A subset of an order is a *cover-preserving* subset if all covering relations in the suborder, induced on this subset, are covering relations in the original order. Thus, P is N-free if there does not exist a, b, c, d in P with covering relations $a \prec c$, $b \prec c$, $b \prec d$, and no other comparabilities on the set $\{a, b, c, d\}$. All series-parallel orders are N-free. The converse however, is not true — see, for example, Figure 1.7. Given any order which contains an N, it is possible to produce an N-free order by “subdividing” all edges. That is, if $a \prec b$, add a new element ab with comparabilities $a \prec ab \prec b$, plus all other relations required by transitivity. The resulting order, while N-free, is still not series-parallel.

A *down set* or *ideal* in an order P , is a suborder D of P such that, if $x \in D$ and $y \prec x$, then $y \in D$. Similarly, $U \subset P$ is an *up set* if $x \in U$

and $y > x$ imply $y \in D$. There is a simple one-to-one correspondence between the up sets and ideals of an order: U is an up set of P if and only if $D = P \setminus U$ is an ideal. Suppose that $x \in D = P \setminus U$ and $y < x$. Then $y \in D$, for if not, $y \in U$ and $x > y$ imply $x \in U$. In addition, an up set in P is an ideal in the dual of P .

Given a subset S of P we define the *down set* of this set by

$$\text{down}(S) = \{x \in P : x \leq y, \text{ for some } y \in S\}.$$

The *up set* of S , $\text{up}(S)$, is defined similarly. There is also an obvious correspondence between the antichains and ideals of an order. Given an ideal D of P , $A = \text{max}(D)$ is an antichain in P , and $\text{down}(A)$ is the ideal D , itself.

A useful related idea is that of "initial" and "final" sets in an order. Let P be a width k order with chain decomposition X_1, X_2, \dots, X_k . Let $S \subset P$ with $|S \cap X_i| \leq 1$ for all $i \leq k$: S contains at most one element from each chain in the decomposition of P . The *initial set* of P generated by S is

$$\text{init}(S) = \bigcup_{X_i \cap S \neq \emptyset} \{x \in X_i : x \leq y, \{y\} = X_i \cap S\}.$$

That is, $\text{init}(S)$ consists of the "initial segment" of each chain in the decomposition up to, and including, the element in that chain, from S . If no element of a chain is contained in S , then none of its elements occur in $\text{init}(S)$. The *final set* of P generated by S is defined similarly. It is clear from the definition of initial sets that $\text{init}(S) \subseteq \text{down}(S)$; but in general, the two need not be equal, so $\text{init}(S)$ is not an ideal. However, we give a simple condition which is sufficient for $\text{init}(S)$ to be an ideal.

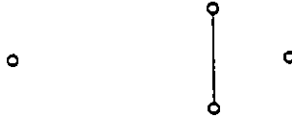


Figure 1.8: Extremal width two orders for $\delta(P) = \frac{1}{3}$

Lemma 1.1 (Initial Set-Ideal) *Let P be a width k order with chain decomposition X_1, X_2, \dots, X_k and let $S \subseteq P$ satisfy $|S \cap X_i| = 1$ ($i \leq k$). Then, $\text{init}(S)$ is an ideal, if for all $a, b \in S$, $a < b$, $a \parallel b$ or $a > b$. If $\text{width } P = 2$ then this condition is also necessary for $\text{init}(S)$ to be an ideal.*

Let $x, y \in P$. Then, x and y are incomparable if and only if $0 < \text{prob}(x < y) < 1$. ($x < y$ if and only if $\text{prob}(x < y) = 1$; $x > y$ if and only if $\text{prob}(x < y) = 0$.) In connection with sorting it is of interest to ask: is there always a pair of elements $x, y \in P$ for which $\text{prob}(x < y)$ is equal to $1/2$, or is at least “close” to $1/2$? Define

$$\delta(P) = \max\{\text{prob}(x < y) : x, y \in P, \text{prob}(x < y) \leq \frac{1}{2}\}.$$

Then the question becomes: is there a constant c , such that for any order P with $\text{width}(P) \geq 2$ (P is not a chain), $\delta(P) \geq c$? The order $2 + 1$, of Figure 1.8, shows that $c \leq \frac{1}{3}$. However, it has been conjectured in [Kislitsyn 68] and [Fredman 76] that for any order P ($\text{width}(P) \geq 2$), $\delta(P) \geq 1/3$. While this general conjecture remains open, some progress has been made.

Theorem 1.10 [Kahn/Saks 84] *For P not a chain, $\delta(P) \geq 3/11$.*

Theorem 1.11 [Linial 84] *For P a width two order, $\delta(P) \geq 1/3$.*

Theorem 1.12 [Aigner 85] *If P is a width two order with $\delta(P) = 1/3$, then P is a linear sum of the orders 1 and $2 + 1$ (of Figure 1.3).*

1.3 Linear Extensions

1.3.1 Finding Linear Extensions

One method of finding a linear extension of an order is *topological sorting*, in which elements are deleted from the order one by one, with the order of selection determining the resulting linear extension. Given the order P , let $P_1 = P$, and choose $x_1 \in \min(P_1)$. Now, let $P_2 = P \setminus \{x_1\}$ and choose $x_2 \in \min(P_2)$. In general, set $P_i = P \setminus \{x_1, x_2, \dots, x_{i-1}\}$ and choose x_i from $\min(P_i)$. Continue until all elements have been selected, producing the linear extension $x_1 < x_2 < \dots < x_n$, where $n = |P|$.

Two important classes of linear extensions which may be formed in this way are “greedy” and “level” linear extensions. The resulting extension is *greedy* if, when choosing x_i , we choose whenever possible, an element of $\min(P)$ which covers x_{i-1} . A *level* or *depth-first* linear extension is formed by always choosing x_i from $\min(P)$ such that $\text{depth}(x_i)$ is maximal.

Example 1.5 Greedy and level linear extensions for the order of Figure 1.9:

- $b < d < f < a < c < e < g < h$ is a greedy linear extension which is not level — both a and c must precede both d and f in a level linear extension.
- $a < b < c < e < d < f < g < h$ is a level linear extension which is not greedy — after e , a greedy linear extension must select g or h .

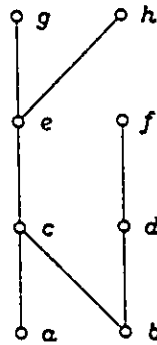


Figure 1.9: Greedy and level linear extensions.

- $a < b < c < d < f < e < g < h$ is neither a greedy nor a level linear extension.
- this order has no linear extensions which are both greedy and level.

Another method of finding a linear extension of an order is *successive partial extensions*. Select a pair of elements x, y which are incomparable in P and assign one of the two possible comparabilities, $x < y$ or $y < x$. Taking the transitive closure gives a partial extension of P . If this extension is not linear, then select another incomparable pair of elements and repeat the process. Much of our work on counting linear extensions is based on this method of producing extensions.

One of the fundamental problems in computer science is that of sorting. Indeed, this is one of the first problems considered by any computer science student. A list of items to be sorted may be viewed as an ordered set, the elements of the order being the items which are to be sorted. Sorting

this list corresponds to finding the “correct” linear extension of the order. Initially, the information is incomplete, the order is not linear. As pairs of the items in the list are compared, new comparabilities are added to the order, extending it. Eventually a linear extension of the original order is achieved: the list is sorted. Thus, the problems of sorting and finding linear extensions of orders are intimately related. In fact, the well known $O(n \log n)$ time bound for optimal comparison sorting algorithms depends on the possible number of linear extensions of an order, corresponding to the number of different possibilities for the sorted list.

Sorting corresponds to finding the “correct” linear extension of an order. The elements are linearly ordered, but that ordering is not known. The task is to determine the ordering. Many scheduling problems involve finding an “optimal” linear extension of an order. A set of “jobs” are to be processed subject to some “precedence constraints”, an order on the jobs. The order is not linear, but the problem dictates that the jobs must be totally ordered. Any linear extension yields a valid schedule, but the object is to find an extension which is optimal with respect to some measure. The difficulty of finding an optimal schedule depends on the number of linear extensions.

1.3.2 Linear Extension Count

The problem of determining the number of linear extensions of an order is a very important one. We focus our attention on this problem — determining the number of extensions, either by direct counting, clever calculation or approximation.

One method of finding and counting linear extensions involves “dynamic

programming", based on topological sorting. We enumerate the extensions by the recursive formula

$$e(P) = \sum_{x \in \max(P)} e(P \setminus x). \quad (1.1)$$

For this method to be computationally practical, the numbers of extensions of the suborders must be recorded efficiently, to avoid recalculating them repeatedly.

Theorem 1.13 [Steiner 90] *If $\text{width}(P) = k$, then $e(P)$ can be calculated in time $O(n^{k+1})$ using $O(n^k)$ space, where $n = |P|$.*

The algorithm which does this finds the set of all ideals of the order, then calculates the number of linear extensions for each ideal. For width two orders, this may be accomplished more efficiently.

Theorem 1.14 [Atkinson/Chang 87] *If $\text{width } P = 2$, then $e(P)$ can be calculated in time $O(n^2)$ using $O(n)$ space, where $n = |P|$.*

The greater efficiency in the width two case results from the fact that the ideals of a width two order are more readily identified.

Theorem 1.15 [Atkinson 90] *If the covering graph of P is a tree, then $e(P)$ may be calculated in time $O(n^2)$, where $n = |P|$.*

The problem of approximating the number of linear extensions of an order has also been considered. One method for doing so considers the "order polytope" defined in [Stanley 86]. Let P be an order on $\{1, 2, \dots, n\}$, then the *order polytope* $\mathcal{O}(P)$ is the convex polytope in \mathbb{R}^n defined by,

$$\mathcal{O}(P) = \{(x_1, x_2, \dots, x_n) \in [0, 1]^n : x_i \leq x_j \text{ whenever } i < j\}.$$

Theorem 1.16 [Stanley 86]

$$\text{volume } \mathcal{O}(P) = \frac{e(P)}{n!}.$$

Thus, $e(P)$ can be calculated by computing the volume of this convex polytope. Unfortunately, Theorem 1.3 [Brightwell/Winkler 91] implies that computing this volume is difficult. However, an efficient method for approximating the volume of this polytope would yield an efficient method of approximating the number of linear extensions of P . This problem has been considered by a number of people.

Theorem 1.17 [Dyer/Frieze/Kannan 89] *Given an n -element order P , and rational numbers $\epsilon, \beta > 0$: $e(P)$ can be approximated with probability at least $1 - \beta$ that the approximation is within $\epsilon e(P)$ of the correct value. This can be done in time polynomial in $n, 1/\epsilon$ and $\log(1/\beta)$.*

The degree of the polynomial in this approximation is quite large, but some improvements have been made, for example [Lovász/Simonovits 90] and [Karzanov/Khachiyan 91]. However, none of the resulting algorithms are considered practical for approximating $e(P)$.

The number of linear extensions of a series-parallel order is easily calculated using the reduction formulas for linear and disjoint sums, given in Example 1.4. That is, assuming that the series-parallel decomposition of the order is known. However, in general, it must first be determined if the order is series-parallel, and a decomposition found, in order to calculate the number of extensions. This has been done efficiently for "simple fold" orders, a small subset of series-parallel orders. An order is a *simple fold* order if it can be produced from a chain by repeated applications of replacing the

interior of a chain by two or more chains; that is, P is a simple fold order if it is of the form

$$P = 1 * (P_1 + P_2 + \dots + P_k) * 1.$$

where each P_i is either a chain or a simple fold order, possibly with the top or bottom elements removed.

Theorem 1.18 [Miller/Stockton 90] *The number of linear extensions of an n -element simple fold order can be calculated in time $O(n^2)$.*

If an order is not a simple fold order, the algorithm will approximate the number of extensions of the order — with the same time complexity — however, the possible error in this approximation is unknown.

In another direction, numerous people have considered the problem of bounding the number of linear extensions, or relating the number to some other value. The following is an extension of equation (1.1),

Theorem 1.19 [Edelman/Hibi/Stanley 89] *If A is an antichain cutset in P , then*

$$e(P) = \sum_{x \in A} e(P \setminus x).$$

This is further extended by,

Theorem 1.20 [Sidorenko 92] *Let A, B be the families of antichains and cutsets, respectively, in P . Then*

$$e(P) = \max_{A \in \mathcal{A}} \sum_{x \in A} e(P \setminus x) = \min_{A \in \mathcal{B}} \sum_{x \in A} e(P \setminus x),$$

with the minimum and maximum occurring precisely when $A \in \mathcal{A} \cap \mathcal{B}$.

Another result in the same paper relates the number of linear extensions of an order to the number of extensions of a complement, if one exists. (It is known that the number of linear extensions is a comparability graph invariant, so all orders complementary to a given order have the same number of linear extensions.)

Theorem 1.21 [Sidorenko 92] *For a pair of complementary n -element orders P, \bar{P} ,*

$$e(P)e(\bar{P}) \geq n!,$$

with equality if and only if P is series-parallel.

The following inequality results from a study of functions defined on the comparability graphs of orders.

Theorem 1.22 [Sidorenko 92] *Let P_1, P_2, \dots, P_k be orders defined on the same set of elements as P , such that the incomparability graphs of the P_i cover P . Then,*

$$e(P_1)e(P_2)\cdots e(P_k) \geq e(P).$$

Chapter 2

Main Results

Our main focus is linear extensions of ordered sets — determining their number and related ideas.

2.1 Linear Extensions

2.1.1 Approximation

We approximate the number of linear extensions of an order by considering “critical” suborders. The behaviour of these approximations is understood for series-parallel orders: successive approximations give a sequence of upper and lower bounds for the number of linear extensions, and eventually, the exact number. For other orders, the answer is not so clear. However, we believe that the approximations behave in a similar manner.

For each critical order there is a corresponding value, its coefficient. The approximations are made by counting the occurrences of the appropriate critical orders and taking the product of the corresponding coefficients raised to this power.

Theorem 2.1 (Alternating Approximations) For a series-parallel order P ,

$$\prod_{|Q| \leq 2i+1, Q \subseteq P} C_Q^{N_Q} \leq e(P) \leq \prod_{|Q| \leq 2i, Q \subseteq P} C_Q^{N_Q} \quad Q \text{ critical,}$$

with equality on the left-hand side if and only if P is a linear sum of critical orders with $2i+1$ or fewer elements, and equality on the right-hand side if and only if P is a linear sum of critical orders with $2i$ or fewer elements.

We conjecture that this result holds for all orders, not just those which are series-parallel.

Let N_2 and N_3 denote the numbers of two- and three-element critical orders in P , respectively. The following has been proved for all orders,

Theorem 2.2 (Upper and Lower Bound) For an order P ,

$$2^{N_2} \left(\frac{3}{4}\right)^{N_3} \leq e(P) \leq 2^{N_2},$$

with equality on the left-hand side if and only if P is a linear sum of one-, two- and three-element critical orders, and equality on the right-hand side if and only if P is a linear sum of one- and two-element critical orders.

We also prove several related results about the critical orders and their coefficients.

Theorem 2.3 (Disconnected Coefficients) A disconnected n -element order is critical, with coefficient

$$C_n = \frac{n}{\prod_{k=2}^{n-1} C_k^{\binom{n-1}{k}}} = \prod_{k=2}^n k^{\binom{n-1}{k-1} (-1)^{n-k}}.$$

Theorem 2.4 (Coefficient Sequence) *The sequence (C_n) of coefficients of the disconnected critical orders satisfies*

- (i) $C_{2n} > 1, C_{2n+1} < 1,$
- (ii) (C_n) converges to 1,
- (iii) $C_{2n}C_{2n+1} > 1, C_{2n+1}C_{2n+2} < 1.$
- (iv) (C_{2n}) is a strictly decreasing sequence,
- (v) (C_{2n+1}) is a strictly increasing sequence.

2.1.2 Enumeration

One of the fundamental ideas used in proving the first step of the approximation gives rise to a method for counting the number of linear extensions of width two orders. Let P be a width two order with chain decomposition X, Y . On the grid $X \times Y$, we draw a graph, whose vertices correspond to the incomparable pairs in P , and whose edges correspond to certain three-element critical orders in P . The linear extensions of P correspond to certain labellings of the vertices of this graph. These labellings are easily enumerated, yielding the number of linear extensions of the order.

Theorem 2.5 (Width Two) *The number of linear extensions of an n -element, width two order can be calculated in time $O(n^2)$.*

This is of course, not a new result. This is the bound achieved in [Atkinson/Chang 87] (Theorem 1.14). In fact, our algorithm for computing the number of linear extensions is very similar to the Atkinson/Chang algorithm. However, the graphs which we use to represent the order provide

more insight into properties of the order related to linear extensions. Indeed, consideration of the development of our algorithm leads to a better understanding of linear extensions of width two orders. This in turn yields an extension of our algorithm to width k orders, which is better than existing algorithms. (c.f. Theorem 1.13)

Theorem 2.6 (Width k) *The number of linear extensions of an n -element width k order can be calculated in time $O(n^k)$.*

Our algorithm for width two orders computes the numbers of linear extensions of all initial sets in P . There are numerous applications of this information.

Let $X = x_1, x_2, \dots, x_p$ and $Y = y_1, y_2, \dots, y_q$ be the chain decomposition of P . (The construction of the graph and enumeration of labellings effectively add a pair of elements, x_0, y_0 , below all other elements of the order. To avoid changing the number of extensions, x_0 and y_0 are necessarily comparable. However, the exact relation between them is unimportant and is not specified.) We prove the following result about the numbers of linear extensions of certain sequences of initial sets.

Theorem 2.7 (Unimodal Sequence) *For an indecomposable width two ordered set P , and $1 < k < |P|$, the sequence*

$$e(\text{init}(x_0 y_k)), e(\text{init}(x_1 y_{k-1})), e(\text{init}(x_2 y_{k-2})), \dots, e(\text{init}(x_k y_0)),$$

is a unimodal sequence which is strictly increasing then strictly decreasing.

If $k > p, q$ then some of the initial or final terms of the sequence are not well defined and are omitted.

More information may be gained by repeating the linear extension calculation on the dual of P , yielding the numbers of linear extensions of the final sets in P . (This, of course, effectively adds another pair of elements x_{p+1}, y_{q+1} , above everything else in the order.) Knowing the numbers of linear extensions of the initial and final sets in P allows us to efficiently calculate the probabilities that certain comparisons or covering relations occur in an extension of P . (In fact, the calculations yield numbers of linear extensions, but probabilities are readily calculated as the number of linear extensions of the order is known.)

Theorem 2.8 (Probabilities) *For x_i incomparable to y_j in P :*

- (i) $e(x_i < y_j) \geq e(\text{init}(x_i, y_{j-1}))e(\text{final}(x_{i+1}, y_j))$ with equality if and only if x_{i+1} and y_{j-1} are comparable;
- (ii) $e(y_j < x_i) \geq e(\text{init}(x_{i-1}, y_j))e(\text{final}(x_i, y_{j+1}))$ with equality if and only if x_{i-1} and y_{j+1} are comparable;
- (iii) $e(x_i < y_j) = e(y_j < x_i) = e(\text{init}(x_{i-1}, y_{j-1}))e(\text{final}(x_{i+1}, y_{j+1}))$;
- (iv) $e(x_{i-1} < y_j < x_i) = e(\text{init}(x_{i-1}, y_{j-1}))e(\text{final}(x_i, y_{j+1}))$;
- (v) $e(y_{j-1} < x_i < y_j) = e(\text{init}(x_{i-1}, y_{j-1}))e(\text{final}(x_{i+1}, y_j))$.

Using the results of the previous two theorems and considering some initial part of the graph of a width two order we give a shorter proof of the characterization (Theorem 1.12, [Aigner, 85]) of those width two orders P , for which $\delta(P) = 1/3$.

2.2 Majority Cycles

It has been known for some time that there are orders which contain linear extension majority cycles (c.f. Theorem 1.4). Many different classes of orders are known to contain majority cycles. We further this knowledge by proving the following.

Theorem 2.9 (Majority Cycle) *There are height two orders containing linear extension majority cycles.*

2.3 Scheduling

In a “flow shop” problem, a set of “jobs” are to be processed by a set of “machines”, possibly subject to some precedence constraints. Each job must be processed exactly once by each machine, passing through the machines in the order M_1, M_2, \dots, M_m . For each pair of a job x and a machine M_i , $t(x, i)$ is the “processing time” required by the machine to process the job. The objective is to find a “schedule” which is “optimal” for a given “measure of performance”. The processing order on any machine is a linear ordering of the jobs. If all machines process the jobs in the same order the schedule is called a permutation schedule.

Theorem 2.10 (Optimal Permutation) *There is an optimal permutation schedule for the m -machine flow shop with precedence constraints, provided that*

$$t(x, i) = t(y, i),$$

for $i = 1, 2, \dots, m$, whenever x and y are incomparable.

The requirement that incomparable jobs have identical processing times is quite restrictive. However, the scarcity of general results for such problems justifies such a restriction. If the measure of performance is "symmetric" then all permutation schedules are optimal. We also show how this result can be used to find upper or lower bounds on optimal schedules in flow shop problems not satisfying this condition on the processing times.

2.4 Genus and Diagram Invariance

We relate the genus of an ordered set to the genus of its covering graph, by means of a "lifting" construction, proving,

Theorem 2.11 (Genus) *The genus of any ordered set equals the genus of its covering graph.*

An immediate consequence of this theorem is the following.

Theorem 2.12 (Diagram Invariant) *The genus of an ordered set is a diagram invariant.*

This is the first, and only, nontrivial example of a diagram invariant.

2.5 Cataloguing Orders

We develop and implement an algorithm to generate a list of the n -element orders having no "irreducible" elements. This algorithm is then applied to the general problem of counting all n -element orders. The algorithm is based, in part, on an idea presented in [Colburn/Read 79]. We present some computational results and discuss possible applications of such an algorithm.

Chapter 3

Approximating Linear Extension Count

We develop a method for approximating the number of linear extensions of an order. Approximations are made by counting the occurrences of suborders belonging to a special class of orders, which we call *critical orders*. The critical orders will be defined as the approximations are developed. Each critical order has a corresponding value, called its *coefficient*. An approximation is made by counting the occurrences of the appropriate critical orders, then taking the product of the corresponding coefficients raised to this power.

One method for finding linear extensions of an order is successive partial extensions. Start with an order P . If P is a linear order then it has only one linear extension, namely, itself. Otherwise, there is a pair of elements x, y , incomparable in P . There are two possible comparisons which may be assigned to this pair, $x < y$ or $y < x$. Choose one, say $x < y$, assign it, and take the transitive closure, producing the order $P_1 = P(x < y)$, an

extension of P . If P_1 is a linear order, then it is a linear extension of P , if not, then P_1 contains a pair of incomparable elements x_1, y_1 . Assign one of the two possible comparisons to this pair of elements, say $x_1 < y_1$. Taking the transitive closure yields the order $P_2 = P_1(x_1 < y_1) = P(x < y, x_1 < y_1)$. If P_2 is not a linear order, then this process continues. Eventually, this results in a linear order, a linear extension of P .

In this procedure, the number of incomparable pairs determines, at least in part, the number of steps required to arrive at a linear extension. Surely, the number of incomparable pairs in an order must give some measure of the number of linear extensions of that order. For an n -element order, the minimum possible number of extensions is one, occurring when the order is a chain, having no incomparable pairs. The maximum number of extensions, $n!$ is achieved by an antichain, with all pairs of elements incomparable.

3.1 Upper Bound

We try to approximate the number of linear extensions of an order by counting the number of incomparable pairs in the order. Let N_2 be the number of incomparable pairs in P , the number of two-element antichains. Since each partial extension of the order introduces at least one more comparability, a linear extension of P is produced in N_2 or fewer steps. When assigning comparisons there are always two choices, so, by a simple combinatorial argument

$$e(P) \leq 2^{N_2}.$$

We define the two-element antichain to be the two-element critical order, its coefficient is $C_2 = 2$.

This inequality is seen as a special case of Theorem 1.22, [Sidorenko 92]. Let x and y be incomparable in P and let $P_{x,y}$ be an order, on the same set of elements as P , in which x, y is the only incomparable pair. The incomparability graph of P is covered by the incomparability graphs of the $P_{x,y}$ (x, y ranging over all incomparable pairs of P). The theorem then states that

$$e(P) \leq \prod_{x \parallel y} e(P_{x,y}).$$

Since $e(P_{x,y}) = 2$ for all incomparable x, y in P , and there are N_2 such orders, it immediately follows that $e(P) \leq 2^{N_2}$.

Our "second" approximation is

$$e(P) \approx A_2(P) = 2^{N_2}.$$

This is called the second approximation both for completeness and for convenience. It is useful to define the "first" approximation to be $A_1(P) = 1$. A singleton is a one-element critical order, having coefficient $C_1 = 1$. While this was not our initial approximation, it is easily seen as a "logical", if somewhat trivial, first step on which to build further approximations. This first approximation satisfies $A_1(P) = 1 \leq e(P)$, with equality if and only if P is a linear order; P contains no incomparable pairs, the two-element critical orders.

We now consider the problem of determining when the second approximation is exact. For $e(P) = 2^{N_2}$ to occur, it must be possible to assign all possible combinations of comparisons on the incomparable pairs of P . There are no comparisons "forced" by transitivity. The incomparable pairs of P are disjoint and are therefore, independent. This is the case if P is



Figure 3.1: Three-element critical orders

a linear sum of singletons and two-element antichains, the one- and two-element critical orders. Otherwise, P contains two incomparable pairs with a common element. In this case, P must contain an indecomposable component with at least three elements. Therefore, P contains a three-element antichain 3, or the order $2 + 1$ (see Figure 3.1), as a suborder. These are the three-element critical orders.

We examine why the approximation fails for the order $2 + 1$. For this order, the approximated number of extensions is four; the actual number is only three. Suppose that the order is defined on the set $\{x, y, z\}$, with $x < y$. Assigning the comparison $z < x$ forces, via transitivity, $z < y$; it is no longer possible to assign $z > y$. Similarly, assigning $z > y$ leaves no choice but to assign $z > x$. Thus, it is not possible to assign both of the comparisons $z < x$ and $y < z$. However, this is the only assignment which is not possible. The other (three) assignments are all possible, they yield the three linear extensions of this order. This is why the actual number of linear extensions of $2+1$ is only $3/4$ of the approximated value. Turning our attention to the three-element antichain, we see that it has six linear extensions, our approximated value is eight. Again, the actual number of extensions is only $3/4$ of the approximated value. In particular, if P contains either of the three-element critical orders, then the approximation

exceeds the actual number of linear extensions. The preceding discussion proves the following.

Theorem 3.1 (Upper Bound) *For a finite order P ,*

$$e(P) \leq 2^{N_3},$$

with equality if and only if P is a linear sum of singletons and two-element antichains.

3.2 Lower Bound

We now attempt to "correct" the (second) approximation. The approximation is refined, by considering the "error" resulting from the presence of the three-element critical orders. This yields a third approximation. The refinement is based on the ratio between the actual number of extensions of these critical orders and the approximated values. For both of the three-element critical orders, $1/4$ of the approximated value represents assignments of comparisons which cannot be made as they would violate transitivity. (cf. Example 3.1) Due to the (multiplicative) nature of the approximation, we expect that a similar result holds for any order containing one of the three-element critical orders as a suborder. That is, if P contains a three-element critical order, then $1/4$ of the approximation for $e(P)$ represents assignments of comparisons which violate transitivity, on this three-element critical order. A similar error is expected for each occurrence of one of the three-element critical orders in P . Therefore, the coefficient of the three-element critical orders is defined to be $C_3 = 3/4$. If N_3 is the number of occurrences of either of the three-element critical

orders in P , then the third approximation is

$$e(P) \approx A_3(P) = 2^{N_2} \left(\frac{3}{4}\right)^{N_3}.$$

Taking the first approximation to be $A_1(P) = 1$, the second approximation can be derived from the first one, in the same way that the third approximation is derived from the second. (This is, of course, working “backwards” since the second approximation must necessarily be defined before the third. However, this development of the second and third approximations will be followed for all subsequent approximations.) The first approximation is exact only if P is a chain, otherwise P contains a two-element antichain. The actual number of linear extensions of this antichain is two, twice the (first) approximation of one. To “correct” the first approximation we multiply it by 2^{N_2} , the “error” in the initial approximation (on a two-element antichain) raised to the power of the number of two-element antichains. Thus, the second approximation is

$$A_2(P) = A_1(P)2^{N_2} = 2^{N_2}.$$

Similarly, the third approximation is

$$A_3(P) = A_2(P) \left(\frac{3}{4}\right)^{N_3} = 2^{N_2} \left(\frac{3}{4}\right)^{N_3}.$$

The definition of A_3 ensures that the third approximation is exact for either of the three-element critical orders. If P does not contain either of the three-element critical orders, then $A_2(P) = e(P)$. In this case, the third approximation, which is the same as the second one, is exact.

All critical orders encountered so far are indecomposable. In fact, we will show that all critical orders are indecomposable. Thus, all critical

orders are contained within indecomposable components. Therefore, the approximations act independently on the indecomposable components of an order. In effect, a separate approximation is made for each component; these approximations are then multiplied together. This is exactly the way that linear extensions behave with respect to the indecomposable components: the number of linear extensions of the order is the product of the numbers of linear extensions of the components. As a result, the third approximation is exact for any order which is a linear sum of one-, two-, or three-element critical orders.

To evaluate the third approximation, we find it useful to adopt a different view of the approximation process. The second approximation may be viewed in a very concrete way, providing a clearer understanding of the approximation process. While this is not possible for subsequent approximations, a clear understanding of the reasoning involved in the initial approximations leads to a better understanding of later approximations.

Thus far, the approximation process was viewed as a dynamic process: a comparison was assigned to an incomparable pair of P ; the transitive closure was found, yielding a partial extension; the process continued, using this partial extension. This process is adaptive, changes made to the order are considered in subsequent operations. We now view the process as one which is not adaptive. Given the order P , (independently) assign comparisons on all the incomparable pairs, without regard to transitivity constraints. There are 2^{N_2} possible combinations of assignments: 2^{N_2} purported linear extensions of P . The objective, now, is to decide which of these assignments are "valid"; that is, which ones produce linear extensions,

42 CHAPTER 3. APPROXIMATING LINEAR EXTENSION COUNT

and which ones do not. See Example 3.1 for an application of this process.

Example 3.1 Possible assignments of comparisons for the three-element critical orders.

$3 = (\{x, y, z\}, \emptyset)$				$2 + 1 = (\{x, y, z\}, \{x < y\})$		
$x \dots y$	$x \dots z$	$y \dots z$	result	$x \dots z$	$y \dots z$	result
<	<	<	$x < y < z$	<	<	$x < y < z$
<	<	>	$x < z < y$	<	>	$x < z < y$
<	>	<	invalid	>	<	invalid
<	>	>	$z < x < y$	>	>	$z < x < y$
>	<	<	$y < x < z$			
>	<	>	invalid			
>	>	<	$y < z < x$			
>	>	>	$z < x < y$			

Example 3.2 Possible assignments of comparisons for the order $2+2=$
 $(\{x, y, u, v\}, \{x < y, u < v\})$.

$x \dots u$	$x \dots v$	$y \dots u$	$y \dots v$	result/intransitive on
<	<	<	<	$x < y < u < v$
<	<	<	>	$\{y, u, v\}$
<	<	>	<	$x < u < y < v$
<	<	>	>	$x < u < v < y$
<	>	<	<	$\{x, y, v\}, \{x, u, v\}$
<	>	<	>	$\{x, u, v\}, \{y, u, v\}$
<	>	>	<	$\{x, y, v\}, \{x, u, v\}$
<	>	>	>	$\{x, u, v\}$
>	<	<	<	$\{x, y, u\}$
>	<	<	>	$\{x, y, u\}, \{y, u, v\}$
>	<	>	<	$u < x < y < v$
>	<	>	>	$u < x < v < y$
>	>	<	<	$\{x, y, u\}, \{x, y, v\}$
>	>	<	>	$\{x, y, u\}, \{y, u, v\}$
>	>	>	<	$\{x, y, v\}$
>	>	>	>	$u < v < x < y$

The third approximation is found by repeatedly multiplying the second one by $3/4$. In effect, each time we multiply by $3/4$, we are attempting to remove those assignments which are intransitive on some particular three-element critical order.

Tables of assignments of comparisons demonstrate what happens when there are several occurrences of the three-element critical orders in an order.

If two critical orders do not share an incomparable pair, then they are "independent". This means that the assignments (of comparisons) which are intransitive on each of these critical orders are independent. This does not mean that there is no overlap of the invalid assignments — quite to the contrary, it means that there is overlap, of a very special form. Suppose that the order P contains the three-element critical orders Q_1 and Q_2 , which do not share an incomparable pair. See, for example, $\{x, y, u\}$ and $\{x, y, v\}$ in Example 3.2. In a table of assignments, the columns corresponding to the incomparable pairs of Q_1 and Q_2 are disjoint, and therefore independent. For each assignment of comparisons on the incomparable pairs of Q_1 , all assignments on the incomparable pairs of Q_2 are possible. Thus, when those assignments which are intransitive on Q_1 are removed from the list, exactly $1/4$ of the remaining assignments are intransitive on Q_2 . Equivalently, exactly $1/4$ of those assignments which are intransitive on Q_1 are also intransitive on Q_2 . Therefore, the number of assignments which are transitive on both Q_1 and Q_2 is $2^{N^2}(3/4)^2$. Extending this argument shows that if no pair of three-element critical orders in P share an incomparable pair, then $e(P) = A_3(P)$. If all three-element critical orders are disjoint, then they cannot share any incomparable pairs. This demonstrates (again)

that A_3 is exact if P is a linear sum of one-, two- and three-element critical orders.

If P is not a linear sum of one-, two- and three-element critical orders, then there is a pair of three-element critical orders which are not disjoint. In this case, it is easily seen that there is a pair of three-element critical orders which share an incomparable pair. When there are critical orders sharing incomparable pairs, the situation is much more complicated. The assignments which are intransitive on the orders are no longer independent. Suppose that the critical orders Q_1 and Q_2 share an incomparable pair. Assignments which are intransitive on Q_1 may be more likely to be intransitive on Q_2 , or they may be less likely to be intransitive. Consider Example 3.2 again. Let $Q_1 = \{x, y, v\}$ and $Q_2 = \{x, u, v\}$. In this case, $1/2$ of the assignments which are intransitive on Q_1 are also intransitive on Q_2 . As a result, $2^{N_2}(3/4)^2$ is less than the number of assignments which are transitive on both Q_1 and Q_2 . This is not cause for concern as we expect $A_3(P)$ to be a lower bound for $e(P)$. However, if $Q_1 = \{x, y, v\}$ and $Q_2 = \{y, u, v\}$, then there are no assignments which are transitive on both Q_1 and Q_2 . Therefore, $2^{N_2}(3/4)^2$ is greater than the number of assignments which are transitive on both Q_1 and Q_2 . This inequality is the reverse of what we are trying to prove. However, a study of tables of assignments leads us to conjecture that $A_3(P)$ is a lower bound for $e(P)$. We prove this result, by different means.

Theorem 3.2 (Lower Bound) *For a finite order P ,*

$$2^{N_2} \left(\frac{3}{4}\right)^{N_3} \leq e(P),$$

with equality if and only if P is a linear sum of critical orders with three or fewer elements.

Proof (Lower Bound) The proof is by induction on $|P|$. The result is trivial for orders with three or fewer elements. Assume that it is true for all orders Q with $|Q| < |P|$. The argument requires several special cases which, since they are not disjoint, are treated successively by excluding from the current case those orders covered by previous cases.

Case (i) $P = P_1 * P_2 * \dots * P_m$ is linearly decomposable.

If P is a linear sum of P_1, P_2, \dots, P_m , then $e(P) = e(P_1)e(P_2)\dots e(P_m)$. Since each critical order is contained in some term of the linear sum, $A_3(P) = A_3(P_1)A_3(P_2)\dots A_3(P_m)$. If each P_i is a linear sum of small (three or fewer elements) critical orders, then $A_3(P_i) = e(P_i)$ for all i , and $A_3(P) = e(P)$. Otherwise, there is an i such that P_i is not a linear sum of small critical orders. Then, $A_3(P_i) < e(P_i)$, which implies that $A_3(P) < e(P)$.

The remaining cases remove an element x from P , and apply the induction hypothesis to $P \setminus x$. The change in the approximation when x is removed must be calculated; this is done by counting the number of two- and three-element critical orders containing x . If x is incomparable to k other elements of P , then x is contained in exactly k incomparable pairs of P . There are at least $\binom{k}{2}$ three-element critical orders containing x , since x along with any pair of its incomparables is a critical order. However, not all three-element critical orders containing x need have this form, so, for some $j \geq 0$,

$$A_3(P) = A_3(P \setminus x) 2^k \left(\frac{3}{4}\right)^{\binom{k}{2}+j} \quad (3.1)$$

The following numerical results are used:

$$2 \left(\frac{3}{4}\right)^k < 1 \quad \text{for } k \geq 2.5; \quad (3.2)$$

$$2^k \left(\frac{3}{4}\right)^{\binom{k}{2}} \text{ is decreasing for } k \geq 3. \quad (3.3)$$

The second follows from the first, and the fact that

$$\frac{2^{k+1} \left(\frac{3}{4}\right)^{\binom{k+1}{2}}}{2^k \left(\frac{3}{4}\right)^{\binom{k}{2}}} = 2 \left(\frac{3}{4}\right)^k.$$

Since the remaining cases assume that P is indecomposable, we must show $A_3(P) < e(P)$. We show that $e(P) \geq \alpha A_3(P)$ for some $\alpha > 1$. As $P \setminus x$ may be decomposable, the induction hypothesis only says that $A_3(P \setminus x) \leq A_3(P \setminus x)$.

Case (ii) P has an isolated element x .

If $|P| = n$ then $e(P) = n e(P \setminus x)$, and

$$A_3(P) = A_3(P \setminus x) 2^{n-1} \left(\frac{3}{4}\right)^{\binom{n-1}{2}}.$$

Since

$$2^{n-1} \left(\frac{3}{4}\right)^{\binom{n-1}{2}}$$

is decreasing for $n \geq 4$, and $2^3(3/4)^3 < 4$,

$$2^{n-1} \left(\frac{3}{4}\right)^{\binom{n-1}{2}} < n \quad \text{for any } n \geq 4.$$

Thus

$$e(P) = n e(P \setminus x) \geq n A_3(P \setminus x) = \left(\frac{n}{2^{n-1} \left(\frac{3}{4}\right)^{\binom{n-1}{2}}}\right) A_3(P) > A_3(P).$$

In the remaining cases, $e(P)$ is related to $e(P \setminus x)$ using this form of Theorem 1.20 [Sidorenko 92]: for any antichain A in P ,

$$\sum_{x \in A} e(P \setminus x) \leq e(P). \quad (3.4)$$

For each $x \in A$, we choose α_x such that $A_3(P) \leq \alpha_x A_3(P \setminus x)$. Then, by (3.4) and the induction hypothesis,

$$e(P) \geq \sum_{x \in A} e(P \setminus x) \geq \sum_{x \in A} A_3(P \setminus x) \geq \left(\sum_{x \in A} \frac{1}{\alpha_x} \right) A_3(P).$$

It suffices to show that

$$\sum_{x \in A} \frac{1}{\alpha_x} > 1.$$

Case (iii) $\text{width}(P) \geq 3$.

Let A be an antichain in P with $|A| \geq 3$. If x is any element of A , then x has $k \geq 2$ incomparables. Since x is neither an isolated, nor a splitting element, there exist y and z comparable and incomparable to x , respectively. Moreover, there exist such y and z which are incomparable to each other. For if not, every element which is comparable to x is also comparable to all elements incomparable to x , which implies that P is linearly decomposable. The set, $\{x, y, z\}$ is a critical order, not of the form of x with two of its incomparables. Therefore, x is contained in at least $\binom{k}{2} + 1$ three-element critical orders, and we may choose

$$\alpha_x = 2^k \left(\frac{3}{4} \right)^{\binom{k}{2} + 1}.$$

By (3.3), α_x is decreasing for $k \geq 3$. Then, since $\alpha_x < 3$ for $k = 2, 3$,

$$\sum_{x \in A} \frac{1}{\alpha_x} > \sum_{x \in A} \frac{1}{3} \geq 3 \left(\frac{1}{3} \right) = 1.$$

Case (iv) $\text{width}(P) = 2$.

Since it is assumed that P is indecomposable and has no isolated elements, P has a (unique) decomposition into two chains, both having size two or greater. Let $u < x$ be the top two elements of one chain and $v < y$ the top two elements of the other. Let k and j be the numbers of elements of P which are incomparable to x and y , respectively. There are (at least) two types of critical orders containing x : those consisting of x and a pair of its incomparables; and those consisting of x and y and an element (other than x) which is incomparable to y . Thus, there are at least $\binom{k}{2} + j - 1$ three-element critical orders containing x . Similarly, there are at least $\binom{j}{2} + k - 1$ critical orders containing y . Since P is indecomposable, x and y are incomparable, and while it is possible that $x > v$ or $y > u$, both cannot occur.

Suppose that there are no comparabilities between x and v , and y and u . Therefore, $k, j \geq 2$. In addition, u and v are incomparable, so $\{x, u, v\}$ and $\{y, u, v\}$ are also critical orders, containing x and y respectively. We choose

$$\alpha_x = 2^k \left(\frac{3}{4}\right)^{\binom{k}{2}+j} \quad \text{and} \quad \alpha_y = 2^j \left(\frac{3}{4}\right)^{\binom{j}{2}+k}.$$

Clearly, α_x is decreasing for all j , and, by (3.3), for all $k \geq 3$. Similarly, α_y is decreasing for all k and all $j \geq 3$. Thus, $1/\alpha_x + 1/\alpha_y$ is increasing for $k, j \geq 3$. It is easily verified that $\alpha_x, \alpha_y < 2$ for $k = 2$ or 3 , and $j = 2$ or 3 , so that, for all $j, k \geq 2$,

$$\frac{1}{\alpha_x} + \frac{1}{\alpha_y} > 1.$$

Suppose now that one of $x > v$ or $y > u$ holds, say $y > u$. Then x and

v must be incomparable. In this case $k \geq 2$ and $j = 1$. Since P contains no splitting elements, there must be an element which is incomparable to u . Let z be the maximal such element. (As z is incomparable to u , z must be contained in the same chain as y . It is possible that $z = v$.) Let w be the upper cover of z , from the same chain as z . By definition of z , $w > u$. But $w \not> x$, so $w > u$. If x and z are comparable then $x > z$. Now, $w > z$, $w > u$, $x > u$ and $x > z$, which implies that P is decomposable, a contradiction. Therefore, x and z must be incomparable. In this case, $\{x, u, z\}$ is another three-element critical order which includes x . We choose

$$\alpha_x = 2^k \left(\frac{3}{4}\right)^{\binom{k}{2}+1} \quad \text{and} \quad \alpha_y = 2 \left(\frac{3}{4}\right)^{k-1}.$$

An argument similar to the previous one verifies that, for $k \geq 2$,

$$\frac{1}{\alpha_x} + \frac{1}{\alpha_y} > 1.$$

□

3.3 Alternating Bounds

If the third approximation is not exact, then a linear decomposition of P contains a term with at least four elements. Therefore, P must contain one of the four-element orders of Figure 3.2 (plus the dual of the fourth, which is not self dual). These are the four-element critical orders.

Emulating what was done previously, we compare the actual numbers of linear extensions of these orders with the approximated values. The four-element antichain 4, has 24 linear extensions, but the approximated value

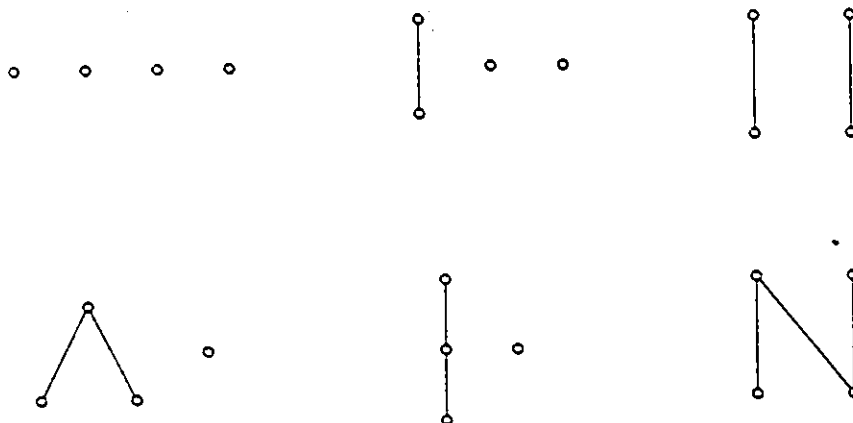


Figure 3.2: Four-element critical orders

is $81/4$. Thus, the ratio of the actual value to the approximated value is

$$\frac{e(4)}{A_3(4)} = \frac{32}{27}.$$

Calculating this ratio for the remaining orders, shows that the ratio is the same for all the orders, except the N. The N has five linear extensions while the approximated value is $9/2$. Thus,

$$\frac{e(N)}{A_3(N)} = \frac{10}{9}.$$

Since the ratio for the N is different, the N must be considered separately. Reasoning similar to that used before, leads us to expect that the approximation is off by a factor of $32/27$ for each occurrence of one of the first five critical orders, and a factor of $10/9$ for each occurrence of N. Define N_4 to be the number of four-element critical orders, other than N, in P . The coefficient of these orders is $C_4 = 32/27$. Let $N_{4,1}$ be the number of N's, the corresponding coefficient is $N_{4,1} = 10/9$. The fourth approximation,

which attempts to correct the third, is

$$A_4(P) = A_3(P) \left(\frac{32}{27}\right)^{N_4} \left(\frac{10}{9}\right)^{N_{4,1}} = 2^{N_2} \left(\frac{3}{4}\right)^{N_3} \left(\frac{32}{27}\right)^{N_4} \left(\frac{10}{9}\right)^{N_{4,1}}.$$

As with the previous approximations, it is easily seen that $A_4(P) = c(P)$ if P is a linear sum of critical orders with four or fewer elements. Otherwise, we expect that the approximation is strictly greater than the number of linear extensions of P .

Conjecture 3.1 (Upper Bound) *For a finite order P ,*

$$c(P) \leq 2^{N_2} \left(\frac{3}{4}\right)^{N_3} \left(\frac{32}{27}\right)^{N_4} \left(\frac{10}{9}\right)^{N_{4,1}},$$

with equality if and only if P is a linear sum of critical orders with four or fewer elements.

Example 3.3 The approximations are illustrated for the two orders in Figure 3.3. Since the five- and six-element critical orders have not been defined yet, only the total number of such critical orders, and the resulting approximations, are given.

Our idea is to make a series of approximations, each a refinement of the previous one, by considering successively larger suborders of the order. We recursively define a class of orders we call *critical orders*, with corresponding *coefficients*, which are used in the approximations. The number of linear extensions of P is approximated by taking the product of the coefficients of the critical orders occurring as suborders of P , counting the number of occurrences. Then new critical orders are defined, based on the

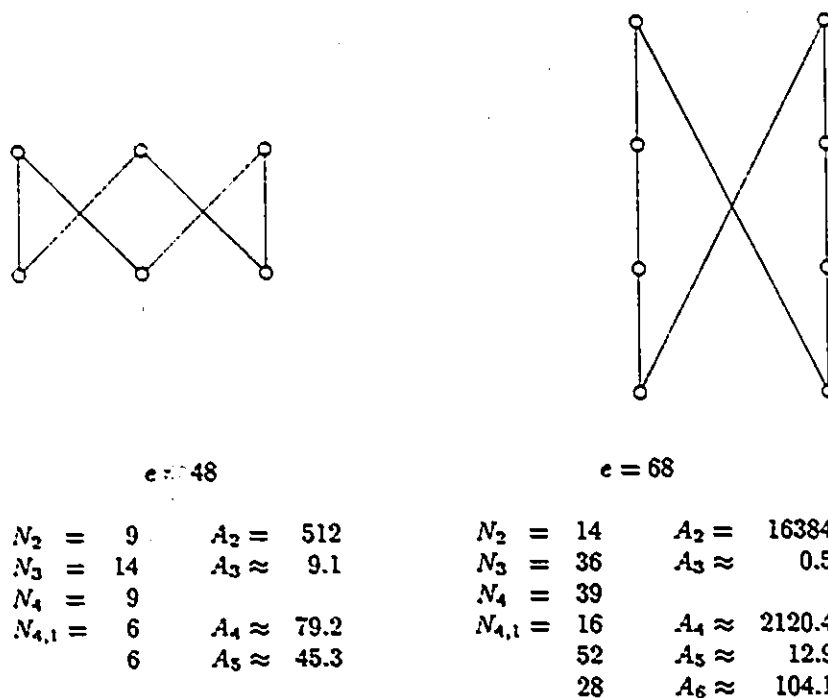


Figure 3.3: Approximation example

orders for which this approximation is “incorrect”; that is, for which the approximation is not exact. The coefficients for these new critical orders are defined to make the necessary “corrections” in the next approximation.

Suppose that the $m - 1$ st approximation, defined as

$$A_{m-1}(P) = \prod_{|Q| \leq m-1, Q \subseteq P} C_Q^{N_Q} \quad Q \text{ critical,}$$

is exact when P is a linear sum of critical orders with $m - 1$ or fewer elements, but is strictly less than the number of linear extensions when P is not such a linear sum. Furthermore, suppose that the critical orders

(with $m-1$ or fewer elements) are precisely the indecomposable orders. Let P be an m -element order. If P is indecomposable then $A_{m-1}(P) < e(P)$. Otherwise, P is a linear sum of orders with $m-1$ or fewer elements, and $A_{m-1}(P) = e(P)$. Therefore, an m -element order is critical, meaning the approximation A_{m-1} is incorrect, if and only if it is indecomposable.

The coefficients of the m -element critical orders are defined so as to make the m th approximation exact on these orders. Let Q be an m -element critical order and define

$$C_Q = \frac{e(Q)}{A_{m-1}(Q)}.$$

Let N_Q be the number of occurrences of Q in P . The m th approximation is defined by

$$A_m(P) = \prod_{|Q| \leq m, Q \subseteq P} C_Q^{N_Q} \quad Q \text{ critical.}$$

Equivalently,

$$A_m(P) = A_{m-1}(P) \prod_{|Q|=m, Q \subseteq P} C_Q^{N_Q} \quad Q \text{ critical.}$$

Then, for an m -element critical order P , $A_m(P) = A_{m-1}(P)C_P = e(P)$. Therefore, if P is any linear sum of critical orders with m or fewer elements, then $A_m(P) = e(P)$. If P is not such a linear sum, we conjecture that $A_m(P)$ is strictly greater than $e(P)$.

For the second and third approximations, when the critical orders were not disjoint, the "corrections" to the approximation were not independent. The result was an "overcorrection". It is reasonable to expect the same behaviour for the m th approximation. Thus, if P is not a linear sum of critical orders with m or fewer elements, there are m element critical orders which overlap, and we expect that $e(P) < A_m(P)$.

In this way we expect to form an alternating sequence of upper and lower bounds for the number of linear extensions.

Conjecture 3.2 (Alternating Approximations) *For a finite order P ,*

$$\prod_{|Q| \leq 2k+1, Q \subseteq P} C_Q^{N_Q} \leq e(P) \leq \prod_{|Q| \leq 2k, Q \subseteq P} C_Q^{N_Q} \quad Q \text{ critical,}$$

with equality on the left-hand side if and only if P is a linear sum of critical orders with $2k + 1$ or fewer elements, and with equality on the right-hand side if and only if P is a linear sum of critical orders with $2k$ or fewer elements.

We are unable to prove this conjecture. It is easily shown that equality holds when an order is a linear sum, of the required form. The difficulty lies in proving that, when the order is not a linear sum, equality fails, and the inequality is in the correct direction. However, we can prove the result for the special case of series-parallel orders. Although calculating the number of linear extensions is easy for series-parallel orders (using the reduction formulas for computing the number of linear extensions of linear and disjoint sums), the fact that the approximation works on these orders supports the general conjecture.

Theorem 3.3 (Alternating Approximations) *For a series-parallel order P ,*

$$\prod_{|Q| \leq 2k+1, Q \subseteq P} C_Q^{N_Q} \leq e(P) \leq \prod_{|Q| \leq 2k, Q \subseteq P} C_Q^{N_Q} \quad Q \text{ critical,}$$

with equality on the left-hand side if and only if P is a linear sum of critical orders with $2k + 1$ or fewer elements, and with equality on the right-hand

side if and only if P is a linear sum of critical orders with $2k$ or fewer elements.

Proof (Alternating Approximations) The proof is by induction on m . The proof requires the additional fact that an m -element series-parallel order is critical if and only if it is linearly indecomposable. That is, an m -element series-parallel order is either critical, or is a linear sum of smaller (series-parallel) critical orders. This fact is easily proved using the induction hypothesis. An m -element order is critical if and only if the approximation A_{m-1} is not exact, which occurs if and only if the order is not a linear sum of critical orders of size $m-1$ or smaller.

The result is true for $m = 2, 3$. Assuming that $m > 3$, the result for A_m is proved by induction on $n = |P|$. If P is a linear sum of critical orders with m or fewer elements, which is certainly the case if $n \leq m$, then it is trivial that $e(P) = A_m(P)$. Suppose that P is not a linear sum of small (m or fewer elements) critical orders. Then, in particular, $n > m$. Suppose also that we have proved the result for all orders smaller, in size, than P . We must show that $e(P) < A_m(P)$ for even m , and $e(P) > A_m(P)$ for odd m . To simplify the argument we consider the case when m is even; the case for odd m is similar.

Since P is series-parallel, it is a linear or disjoint sum of the series-parallel orders P_1, P_2 . The induction hypothesis applies to P_1 and P_2 . Therefore, $e(P_1) \leq A_m(P_1)$ and $e(P_2) \leq A_m(P_2)$.

If $P = P_1 * P_2$ then $e(P) = e(P_1)e(P_2)$ and $A_m(P) = A_m(P_1)A_m(P_2)$. Moreover, since P is not a linear sum of small critical orders, it cannot

happen that both P_1 and P_2 are linear sums of small critical orders. Thus, $e(P_1) < A_m(P_1)$ or $e(P_2) < A_m(P_2)$, and so $e(P) < A_m(P)$.

Otherwise, $P = P_1 + P_2$. Let $k = |P_1|$, so $|P_2| = n - k$. (Without loss of generality we may assume that $k \leq n/2$.) Then,

$$e(P) = \binom{n}{k} e(P_1) e(P_2).$$

There are three types of critical orders contained in P : those contained in P_1 ; those contained in P_2 ; those contained partly in P_1 and partly in P_2 . Any suborder of the last type is disconnected. We will show (Theorem 3.4) that all disconnected orders are critical, and all l -element critical orders have the same coefficient C_l , which depends only upon l . Therefore, all suborders contained partly in P_1 and partly in P_2 are critical. There are $\binom{n}{l} - \binom{k}{l} - \binom{n-k}{l}$ such l -element critical orders. Therefore,

$$A_m(P) = A_m(P_1) A_m(P_2) \prod_{l=2}^m C_l^{\binom{n}{l} - \binom{k}{l} - \binom{n-k}{l}}.$$

Then

$$\frac{A_m(P)}{e(P)} \geq \frac{\prod_{l=2}^m C_l^{\binom{n}{l} - \binom{k}{l} - \binom{n-k}{l}}}{\binom{n}{k}}.$$

It is sufficient to show that this is greater than 1, or equivalently

$$\sum_{l=2}^m \left[\binom{n}{l} - \binom{k}{l} - \binom{n-k}{l} \right] \ln C_l - \ln \binom{n}{k} > 0.$$

For odd m , the inequality is reversed. The two cases may be combined by multiplying by a factor of $(-1)^m$. Define

$$f(k) = (-1)^m \sum_{l=2}^m \left[\binom{n}{l} - \binom{k}{l} - \binom{n-k}{l} \right] \ln C_l + (-1)^{m+1} \ln \binom{n}{k}.$$

It is sufficient to show that $f(k) > 0$ for $1 \leq k \leq n/2$. To this end, we show that $f(1) > 0$ and that f is increasing for $k \leq n/2$.

$$\begin{aligned} f(1) &= (-1)^m \sum_{l=2}^m \left[\binom{n}{l} - \binom{1}{l} - \binom{n-1}{l} \right] \ln C_l + (-1)^{m+1} \ln \binom{n}{1} \\ &= (-1)^m \sum_{l=2}^m \binom{n-1}{l-1} \ln C_l + (-1)^{m+1} \ln n. \end{aligned}$$

This value depends on n . Accordingly, we define $g(n) = f(1)$, and show that $g(n) > 0$, for $n > m$. This is accomplished by defining a (finite) sequence of functions, starting from g , with the property that, if a function in the sequence is positive, then the preceding function is increasing, and therefore positive. Showing that the last function is positive we conclude that g is positive.

If $n \leq m$ then $A_m(P) = e(P)$, which implies that $g(n) = 0$, for $n \leq m$. It suffices to show that g is (strictly) increasing for $n > m$; for then, $n > m$ implies $g(n) \geq g(m+1) > g(m) = 0$. Define

$$\Delta g(n) = g(n+1) - g(n).$$

We claim that $\Delta g(n) > 0$ when $n \geq m$, and therefore, $g(n)$ is increasing for $n \geq m$. If $n \leq m-1$ then $\Delta g(n) = 0 - 0 = 0$. Thus, to show that $\Delta g(n) > 0$ for $n > m-1$, it suffices to show that Δg is increasing for $n > m-1$. Define

$$\Delta^2 g(n) = \Delta g(n+1) - \Delta g(n),$$

which we show is greater than 0 when $n > m-2$, proving that $\Delta g(n)$ is increasing, hence greater than 0, for $n \geq m$. This process continues until

we arrive at $\Delta^m g(n)$, which we will show is greater than 0 for all $n \geq 1$. This will prove the desired result for g .

The function g is the sum of a logarithm term and binomial coefficients, and is therefore, a C^∞ function.

Thus,

$$\Delta g(n) = g(n+1) - g(n) = \int_n^{n+1} g'(x_1) dx_1,$$

$$\Delta^2 g(n) = \Delta(\Delta g(n)) = \int_n^{n+1} \Delta(g'(x_1)) dx_1 = \int_n^{n+1} \int_{x_1}^{x_1+1} g''(x_2) dx_2 dx_1,$$

and in general,

$$\Delta^i g(n) = \int_n^{n+1} \int_{x_1}^{x_1+1} \dots \int_{x_{i-1}}^{x_{i-1}+1} g^{(i)}(x_i) dx_i \dots dx_1.$$

The polynomial term in g has degree $m-1$, since the binomial coefficient $\binom{n-1}{l-1}$ is a polynomial in n , of degree $l-1$, and $l \leq m$. Therefore, the polynomial term in g becomes 0 with the m th derivative. Thus, $\Delta^m g(n) = \Delta^m(-1)^{m+1} \ln n$, and

$$\Delta^m g(n) = \int_n^{n+1} \int_{x_1}^{x_1+1} \dots \int_{x_{m-1}}^{x_{m-1}+1} \frac{(m-1)!}{x_m^m} dx_m \dots dx_1.$$

Since $n \geq 1$, it follows that $x_m \geq 1$. Then, $\Delta^m g(n) > 0$, since, at each step, a positive function is being integrated. Therefore, we conclude that all $\Delta^i g(n)$ are greater than 0 for the specified values of n , so $f(1) > 0$ when $n > m$.

We now show that $f(k)$ is increasing for $k \leq n/2$. Suppose that $1 \leq k < n/2$. By a straightforward calculation, $f(k+1) - f(k) = g(n-k) - g(k)$. Since $k < n/2$, $n-k > k$, and as g is increasing, $g(n-k) > g(k)$. Therefore, $f(k+1) > f(k)$; that is, f is increasing. \square

3.4 Critical Orders and Coefficients

The m -element critical orders are defined by the orders for which the approximation A_{m-1} is incorrect. An m -element order Q is critical if $A_{m-1}(Q) \neq e(Q)$. The coefficient of Q is defined as $e(Q)/A_{m-1}(Q)$. Thus, the m -element critical orders are precisely the m -element orders with coefficients different from 1.

It is important to know which orders are critical. Suppose that P is an m -element order. If P is linearly decomposable, then it is a linear sum of orders with $m-1$ or fewer elements. In this case, $A_{m-1}(P) = e(P)$, whence P is not critical. Therefore, critical orders are linearly indecomposable. As stated previously, if the alternating approximation conjecture is true, then the critical orders are precisely the indecomposable orders. In this case, almost all orders are critical, since it is known that, asymptotically, almost all orders are linearly indecomposable, (Theorem 1.8, [Möhring 84]).

We now consider the coefficients of the critical orders. In general, not all critical orders (of a given size) have the same coefficient. However, all disconnected orders are critical, with coefficients depending only on the size of the order.

Theorem 3.4 (Disconnected Coefficients) *A disconnected n -element order is critical, with coefficient*

$$C_n = \frac{n}{\prod_{k=2}^{n-1} C_k^{\binom{n-1}{k}}} = \prod_{k=2}^n k^{\binom{n-1}{k}(-1)^{n-k}}.$$

Proof (Disconnected Coefficients) The proof is by induction on n . The result is true for $n = 2$. Suppose that the result is true for all orders with

$n - 1$ or fewer elements. Let P be an n -element disconnected order. Say $P = P_1 + P_2$, with $m = |P_1|$ and $|P_2| = n - m$. Then

$$e(P) = \binom{n}{m} e(P_1) e(P_2),$$

and $m, n - m \leq n - 1$ implies $A_{n-1}(P_1) = e(P_1)$ and $A_{n-1}(P_2) = e(P_2)$.

There are three types of critical orders in P : those contained in P_1 ; those contained in P_2 ; those contained partly in P_1 and partly in P_2 . In fact, all suborders of the last type are disconnected, and therefore critical, by the induction hypothesis. The number of k -element critical orders of the last type is $\binom{n}{k} - \binom{m}{k} - \binom{n-m}{k}$. Therefore

$$A_{n-1}(P) = A_{n-1}(P_1) A_{n-1}(P_2) \prod_{k=2}^{n-1} C_k^{\binom{n}{k} - \binom{m}{k} - \binom{n-m}{k}}.$$

The coefficient of P is

$$C_P = \frac{e(P)}{A_{n-1}(P)} = \frac{\binom{n}{m}}{\prod_{k=2}^{n-1} C_k^{\binom{n}{k} - \binom{m}{k} - \binom{n-m}{k}}}. \quad (3.5)$$

We note that this coefficient depends only on n and m . That is, the coefficient of P depends only on the sizes of P_1 and P_2 , not their structure. Any other n -element order which is a disjoint sum of two orders of the same size as P_1 and P_2 , will have the same coefficient. In fact, we will show that the value of this coefficient is not dependent upon m , that it is constant, for all m with $1 \leq m \leq n - 1$.

Consider the n -element antichain n . Its coefficient is defined in terms of the number of its linear extensions, and the critical orders that it contains. Since these cannot vary, the antichain has a single, well defined coefficient C_n . However, the antichain can be written as a disjoint sum of orders of

size m and $n - m$, for any m with $1 \leq m \leq n - 1$. Therefore, any value of m ($1 \leq m \leq n - 1$) may be used in equation (3.5), to calculate C_n . Since all values of m must give the same result, the coefficients do not depend on m . In particular, for $m = 1$ we have

$$C_n = \frac{\binom{n}{1}}{\prod_{k=2}^{n-1} C_k^{\binom{n}{k} - \binom{n}{k-1}}} = \frac{n}{\prod_{k=2}^{n-1} C_k^{\binom{n-1}{k-1}}}.$$

Therefore, all disconnected n -element orders have the same coefficient C_n .

The second formula is proved by considering the exponent of k in C_n . We show, by induction on n , that it is $\binom{n-1}{k-1}(-1)^{n-k}$. If $n = k$ then the exponent is 1, which is correct. Suppose now that $n > k$. For all m , with $k \leq m < n$, the induction hypothesis implies that the exponent of k in C_m is $\binom{m-1}{k-1}(-1)^{m-k}$. The exponent of C_m in C_n is $\binom{n-1}{m-1}$. Applying the identity

$$\binom{a}{b} \binom{b}{c} = \binom{a}{c} \binom{a-c}{b-c},$$

the exponent of k in C_n is,

$$\begin{aligned} \sum_{m=k}^{n-1} \binom{n-1}{m-1} \binom{m-1}{k-1} (-1)^{m-k} &= \sum_{m=k}^{n-1} \binom{n-1}{k-1} \binom{n-k}{m-k} (-1)^{m-k} \\ &= \binom{n-1}{k-1} \sum_{l=0}^{n-k-1} \binom{n-k}{l} (-1)^l = \binom{n-1}{k-1} (-1)^{n-k}. \end{aligned}$$

To show that all disconnected n -element orders are critical we need only show that $C_n \neq 1$. Let p be the largest prime number such that $p \leq n$. Then, $p > n/2$, for otherwise there would be a greater prime q , between p and $2p \leq n$. Since $2p > n$, no multiple of p occurs in the product computing C_n . Therefore, the exponent of p in C_n is $\binom{n-1}{p-1}(-1)^{n-p}$. Since

p has nonzero exponent, $C_n \neq 1$, and all n -element disconnected orders are critical. \square

If the general conjecture is correct, then the coefficients of a critical order must be greater than 1, or less than 1, depending on whether the size of the critical order is even or odd, respectively. Furthermore, for the approximations to converge to the correct number of linear extensions, we expect that the coefficients converge to 1, as the size of the critical orders increase. As the coefficient of each connected critical order must be calculated separately, we are unable to prove any results about their values. However, it is possible to study the sequence of coefficients for the disconnected critical orders, which behave as expected.

Theorem 3.5 (Coefficient Sequence) *The sequence (C_n) of coefficients of the disconnected critical orders satisfies*

- (i) $C_{2n} > 1, C_{2n+1} < 1$,
- (ii) (C_n) converges to 1,
- (iii) $C_{2n}C_{2n+1} > 1, C_{2n+1}C_{2n+2} < 1$,
- (iv) (C_{2n}) is a strictly decreasing sequence,
- (v) (C_{2n+1}) is a strictly increasing sequence.

Proof (Coefficient Sequence) To study the sequence (C_n) , we consider the sequence of functions (f_n) , defined by

$$f_n(x) = (-1)^{n+1} \Delta^n \ln x = \sum_{l=0}^n (-1)^{l+1} \binom{n}{l} \ln(x+l) \quad \text{for } x > 0.$$

Then,

$$\begin{aligned}\ln C_n &= \ln \left(\prod_{k=2}^n k^{\binom{n-1}{k-1} (-1)^{n-k}} \right) = \sum_{k=2}^n (-1)^{n-k} \binom{n-1}{k-1} \ln k \\ &= \sum_{k=1}^n (-1)^{n-k} \binom{n-1}{k-1} \ln k = (-1)^n \sum_{l=0}^{n-1} (-1)^{l+1} \binom{n-1}{l} \ln(l+1).\end{aligned}$$

That is,

$$\ln C_n = (-1)^n f_{n-1}(1).$$

In the proof of the Alternating Approximations Theorem, we show that $f_n(x) > 0$ for $x \geq 1$. In fact the same argument proves this is true for all $x > 0$. Thus, for even n , $\ln C_n > 0$, so $C_n > 1$. For odd n , $\ln C_n < 0$, so $C_n < 1$.

To show that (C_n) converges to 1, we show that $(f_n(1))$ converges to 0. By definition of f_n ,

$$f_{n+1}(x) = (-1)\Delta f_n(x) = f_n(x) - f_n(x+1).$$

Then, $f_n(x+1) > 0$ implies that $f_n(x) > f_{n+1}(x)$. Thus, for any x , $(f_n(x))$ is a positive sequence which is decreasing, hence convergent. Therefore, the sequence (f_n) converges pointwise to some function f . We show that f is continuous by showing that the convergence is uniform. It is sufficient to show that the functions f_n are decreasing, since monotonicity and pointwise convergence imply uniform convergence. We show that

$$f'_n(x) = (-1)^{n+1} \Delta^n \frac{1}{x} < 0,$$

by taking the n th derivative and integrating n times, as was done in the proof of the Alternating Approximations Theorem. As

$$\frac{d^n}{dx^n} f'_n(x) = \frac{-n!}{x^{n+1}} < 0,$$

all integrals are negative, and f_n is decreasing. Thus (f_n) converges uniformly to f , which is therefore, continuous.

Now we show that $f(x) = 0$, for all $x > 1$, which, by continuity, implies that $f(1) = 0$. Therefore, $(f_n(1))$ converges to 0. For $x > 0$, $f_n(x+1) = f_n(x) - f_{n+1}(x)$. As $(f_n(x))$ converges, $f_n(x) - f_{n+1}(x)$ approaches 0, so $(f_n(x+1))$ converges to 0. That is, $f(x) = 0$ for $x > 1$. Therefore, $(\ln C_n)$ converges to 0, whence (C_n) converges to 1.

Since $(f_n(1))$ is decreasing, $f_{2n}(1) - f_{2n+1}(1) > 0$. This may be rewritten as $(-1)^n f_n(1) + (-1)^{n+1} f_{n+1}(1) > 0$. Equivalently, $C_{2n} C_{2n+1} > 1$. A similar argument shows that $C_{2n+1} C_{2n+2} < 1$. From $f_{2n}(1) - f_{2n+2}(1) > 0$ we conclude that $C_{2n} > C_{2n+2}$. Thus (C_{2n}) is a strictly decreasing sequence. Similarly (C_{2n+1}) is a strictly increasing sequence. \square

Figure 3.4 shows the five-element critical orders, with their coefficients. Note that all of the coefficients are less than 1.

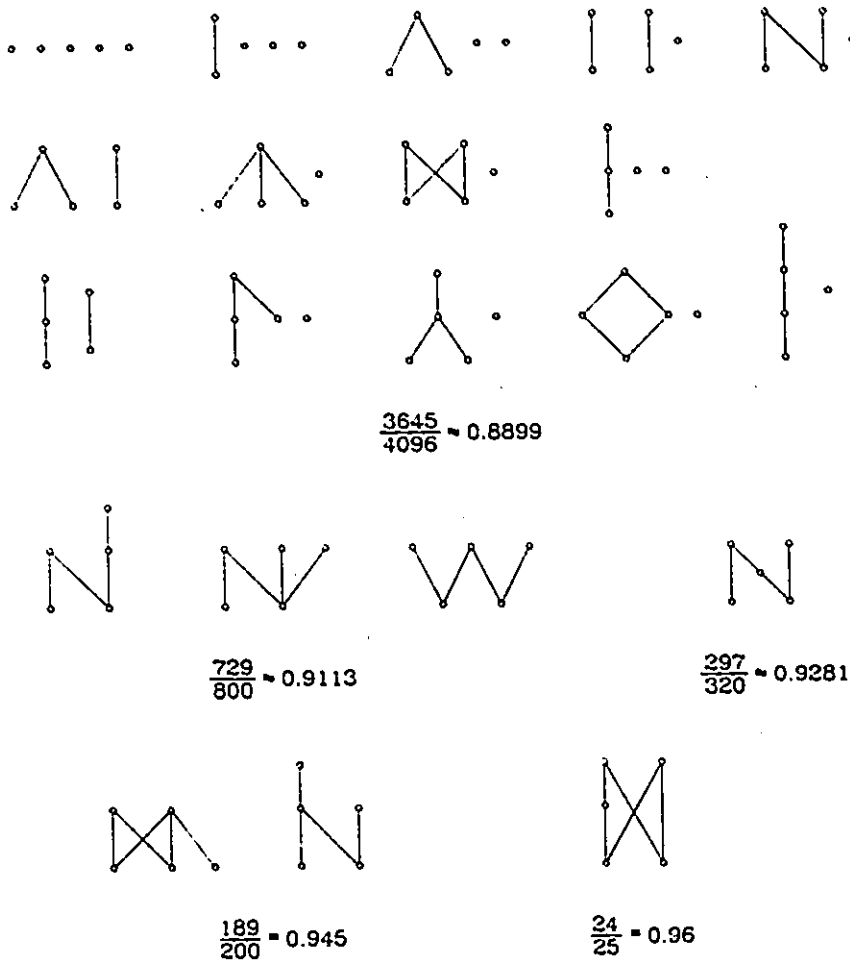


Figure 3.4: Five-element critical orders

Chapter 4

Linear Extensions of Width Two Orders

We devise a novel representation for width two orders, by means of a graph drawn on a grid. This representation yields an efficient algorithm to compute the number of linear extensions of a width two order. This in turn yields an efficient method for calculating probabilities of specific comparabilities occurring in linear extensions of the order. Our graphical representation of the order also yields a new result on unimodal sequences.

In the previous chapter, the number of linear extensions of an order P was approximated by considering the occurrences of critical orders in P . The second approximation $A_2(P)$, corresponds to the total number of assignments of comparisons on the incomparable pairs of P , where such assignments are made without regard to transitivity. We tried to calculate what fraction of these assignments are valid by considering the occurrences of the three-element critical orders in P . No attempt was made at deciding which specific assignments were valid, and which were not. Doing

so would yield the exact number of linear extensions of P . However, this is obviously (computationally) difficult, since we know that determining the number of linear extensions of an order is #P-complete (Theorem 1.3, [Brightwell/Winkler 91]). It may be possible that this problem is simpler for some restricted class of orders. In fact, we show that for width two orders, it is easy to decide which assignments of comparisons correspond to linear extensions of P , and to count their number.

We represent a width two order P by a graph drawn on a rectangular grid, the vertices of the graph corresponding to the incomparable pairs of the order. Labellings of the vertices of the graph correspond to assignments of comparisons on the incomparable pairs of the order. The structure of the graph facilitates identification and enumeration of those labellings corresponding to linear extensions of the order.

An alternative interpretation of this algorithm yields an algorithm for computing the number of linear extensions of orders with width greater than two.

4.1 Graph Construction

Let P be a width two order, with chain decomposition $X = x_1 < x_2 < \dots < x_p$, $Y = y_1 < y_2 < \dots < y_q$. (A width two order has a unique chain decomposition if and only if it is linearly indecomposable. If P is decomposable, any chain decomposition may be used.) The graph G of P , is drawn in the rectangular grid $R = X \times Y$. The elements of X label the horizontal axis, with x_i in position i . The elements of Y label the vertical axis, with y_j in position j . The rectangle R has corners at $(1,1)$

and (p, q) . (The points are identified by Cartesian coordinates, rather than as the elements of a matrix.) Each point in R corresponds to a pair of elements of P , one from each chain. The point corresponding to x, y is at position (i, j) , and is referred to by either notation. Since the graph is drawn in a grid, we may refer to the rows and columns of the graph.

There is a vertex at point xy if and only if x and y are incomparable in P . Edges are drawn between all pairs of vertices at adjacent points in the grid. (See, for example, Figure 4.1.) Two points in the grid are adjacent if they agree in one coordinate and differ by one in the other coordinate. Thus, $(i \pm 1, j)$ and $(i, j \pm 1)$ are all adjacent to (i, j) . If the chains X and Y are exchanged, then a reflection (across the diagonal) of G is constructed. More generally, different chain decompositions yield different graphs, which will be explained later. However, these graphs are all similar, so we construct the graph corresponding to the given chain decomposition, and refer to it as the graph of the order.

To describe the resulting graphs, define the following properties for a graph $G = (V, E)$ drawn on a grid. Call the graph *row increasing* (or equivalently, *column increasing*) if each row is shifted to the right of the rows below it (each column is shifted up from the columns to the left of it). That is, if (f_j, j) and (l_j, j) are the positions of the first and last vertices in the j th row, then for all $j < j'$; $f_j \leq f_{j'}$ and $l_j \leq l_{j'}$. (If (i, b_i) and (i, t_i) are the positions of the bottom and top vertices in the i th column, then for all $i < i'$; $b_i \leq b_{i'}$ and $t_i \leq t_{i'}$.) Equivalently, if the upper left and lower right corners of any rectangle in R are contained in V , then all points in the rectangle must be contained in V . That is, if $(i_1, j_2), (i_2, j_1) \in V$, then

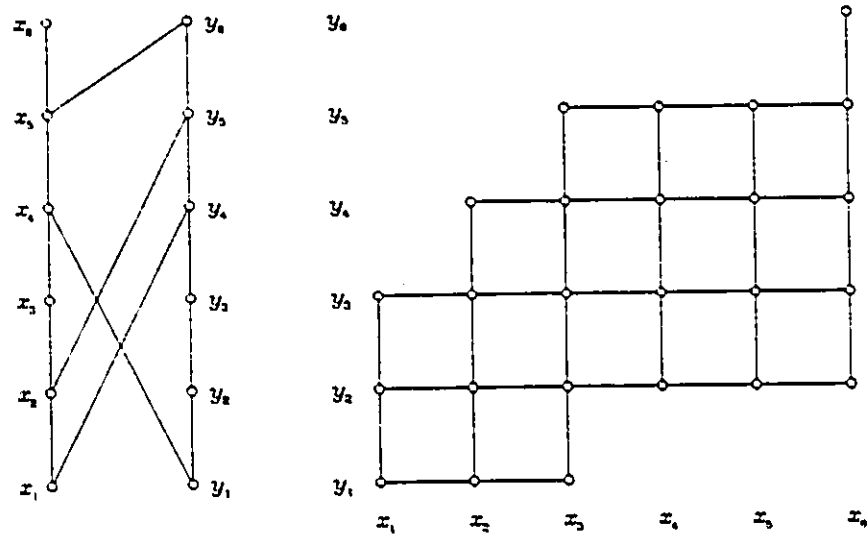


Figure 4.1: A width two order and its graph

$(i_2, j_2) \in V$, for all $i_1 \leq i_2 \leq i_3, j_1 \leq j_2 \leq j_3$. A special case of this occurs when $i_1 = i_3$ or $j_1 = j_3$, in which case the property asserts that there are no gaps between the vertices in any row or column. We call this property *vertex contiguity*.

Call G *grid-edge complete* if all valid grid edges occur in G . That is, if $(i, j), (i + 1, j) \in V$ then $((i, j), (i + 1, j)) \in E$, and if $(i, j), (i, j + 1) \in V$ then $((i, j), (i, j + 1)) \in E$. The graph is connected if and only if $f_{i+1} \leq l_i$, for all i (equivalently $b_{j+1} \leq t_j$ for all j).

Theorem 4.1 *If P is a width two order, then its graph G is row increasing and grid-edge complete. Furthermore, G is connected if and only if P is linearly indecomposable.*

Proof First we show that G is row increasing and grid-edge complete. Let $w_1 \leq w_2 \leq w_3$ in X , and $z_1 \leq z_2 \leq z_3$ in Y . If $w_1 z_3, w_3 z_1 \in V$, then w_1 and z_3 are incomparable, as are w_3 and z_1 . If $w_2 < z_2$, then $w_1 < z_3$. Similarly $w_2 > z_2$ implies that $w_3 > z_1$. Therefore w_2 is incomparable to z_2 , so $w_2 z_2 \in V$. This proves that G is row increasing. By construction, G is grid-edge complete.

Suppose that $w_1 z_1, w_1 z_2 \in V$. Then w_1 incomparable to z_1 , and w_1 incomparable to z_2 , imply that z_1, z_2 and w_1 are all in the same indecomposable component of P . Similarly, if $w_1 z_1, w_2 z_1 \in V$, then w_1, w_2 and z_1 are all in the same (indecomposable) component of P . That is, if a pair of vertices lie in the same row or column of G , then the corresponding elements of P are all in the same component of P . Suppose that xy and $x'y'$ are in the same (connected) component of G . There is a path from xy to $x'y'$, consisting of a series of horizontal and vertical edges. It follows that x, x', y and y' are all contained in the same indecomposable component of P . Therefore, G being connected implies that P is indecomposable.

Now, suppose that G is disconnected. Let xy and $x'y'$ be vertices from two different components of G . Suppose that $x \leq x'$ but $y \geq y'$. Then xy and $x'y'$ are the upper left and lower right corners, respectively, of a rectangle in G . By the row increasing property, G has vertices at all points in this rectangle. This implies that there is a path connecting the two vertices, contradicting the fact that these vertices are from different components of G . A similar contradiction obtains if we assume that $x \geq x'$ and $y \leq y'$. Therefore, either $x < x'$ and $y < y'$, or $x > x'$ and $y > y'$. That is, each component of G is strictly above and to the right of the

previous component. The row increasing property also guarantees that each component has well defined upper right and lower left vertices. Now, let xy be the upper right vertex of one component of G , and $x'y'$ the lower left vertex of the next component. Let u be the upper cover, from X , of x , and v be the upper cover, from Y , of y . (Then $x < u \leq x'$ and $y < v \leq y'$. Equality need not hold in either case.) As G is vertex contiguous, there cannot be any vertices directly above or directly to the right of xy . Therefore, x is comparable to all elements above y , and y is comparable to all elements above x . In particular, $x < v$ and $y < u$. Thus P is decomposable, with the sum occurring between x, y and u, v . \square

In fact, the proof shows not only that there is a correspondence between indecomposable orders and connected graphs, but that there is a correspondence between the indecomposable components of an order P , and the connected components of its graph G . If C is a connected component of G , then C is contained in the rectangle R_C defined by the upper right and lower left vertices of C . Let X_C and Y_C be the subchains of X and Y , respectively, whose product is R_C . Then $X_C \cup Y_C$ is the component of P which corresponds to C . Another chain decomposition of P may be formed by exchanging X_C and Y_C . That is, if $X = X_1 * X_C * X_2$ and $Y = Y_1 * Y_C * Y_2$ are the chains in the decomposition, then $X_1 * Y_C * X_2$ and $Y_1 * X_C * Y_2$ are the chains in another chain decomposition of P . It is possible to swap X_C and Y_C since $X_C \cup Y_C$ is an indecomposable component of P . The graph corresponding to this new decomposition is similar to G , except that it contains the reflection of C , in place of C . (If $|X_C| \neq |Y_C|$ then the number of rows and columns in the graph will change accordingly.)

In fact, given any two graphs for an order, one can be obtained from the other by reflecting components.

It is useful to know which graphs on a grid correspond to width two orders. Suppose that we are searching for a width two order satisfying some property readily identified in its graph. It may be easier to construct a graph with the required property, than to construct such an order. In fact any row increasing, grid-edge complete graph, $G = (V, E)$, on a grid, is the graph of some width two order. For simplicity, we assume that G is connected. If G is disconnected, each connected component corresponds to a term in a linear decomposition, and may be considered separately.

Let R be the rectangle formed by the lower left and upper right vertices of G , and define the chains $X = x_1 < x_2 < \dots < x_p$ and $Y = y_1 < y_2 < \dots < y_q$ such that $R = X \times Y$. The order P is defined by specifying the comparabilities between X and Y . Let $x \in X, y \in Y$. If $xy \in V$ then x and y are incomparable. Otherwise, $xy \notin V$ and x and y are comparable. We must decide which way to assign the comparability. Since G is connected, there is a vertex in each column of R . Thus, there is a vertex either above or below xy . However, vertex contiguity implies that there cannot be vertices both above and below xy , since that would require a vertex at xy . If there are vertices below xy , define $x > y$. Otherwise, there are vertices above xy ; define $x < y$. This defines the relation between any pair of elements of P .

The relations could also have been defined by considering the rows of the graph instead of the columns. Similar arguments show that each row contains at least one vertex, and if there is no vertex at a point xy in a row, then there must be vertices either to the left or right of xy , but not

both. If there is a vertex at xy , then x and y are incomparable, as before. If there is no vertex at xy , then we define $x < y$ if there are vertices to the left of xy , and $x > y$ if there are vertices to the right of xy . These two definitions of the relation are equivalent since the row increasing property implies that if there is no vertex at a point, then there are vertices above the point if and only if there are vertices to the left of the point, and there are vertices below the point if and only if there are vertices to the right of the point.

Proposition 4.1 *If G is a row increasing, grid-edge complete graph, then P is a width two order.*

Proof We must show that the relation as defined is transitive. Suppose that $u < v$ and $v < w$. We prove that $u < w$. There are several cases depending on the distribution of the elements u, v and w in the two chains.

Case (i) u, v, w are in the same chain.

Trivially $u < w$, as the chain is linearly ordered.

Case (ii) u, v are in one chain and w is in the other chain.

Suppose that $u, v \in Y$ and $w \in X$. Since $u < v$ in Y , the vertices above vw , in row w of G , are also above uw . There is no vertex at uw , for otherwise, vertex contiguity would imply that there is a vertex at vw . Therefore $u < w$.

Case (iii) u is in one chain and v, w are in the other chain.

This case is similar to *Case (ii)*.

Case (iv) u, w are in one chain and v is in the other chain.

Say $u, w \in Y$ and $v \in X$. If $w < u$ then the situation is the same as *Case (ii)*, but for w, u, v , instead of u, v, w . This leads to the conclusion $w < v$, a contradiction. Therefore, $u < w$, as u and w are comparable.

Thus, P is a valid order. It is clear from the construction of P that *width* P is no greater than two. However, since there are incomparable pairs in P , the width is exactly two. \square

4.2 Counting Linear Extensions

We now consider the problem of relating 0,1 labellings of the graph of an order to linear extensions of the order, and counting the number of such labellings. A 0,1 labelling of the graph is a labelling of the vertices of the graph as either 0 or 1. The labelling of a vertex specifies the comparison to be assigned to the corresponding incomparable pair. A valid labelling is one which corresponds to some linear extension. Not all labellings of the graph correspond to linear extensions of the order. However, the structure of the graph makes it particularly easy to identify and enumerate the valid labellings.

Let P be a width two order with chain decomposition X, Y , and let G be its graph. Suppose that $x \in X$ and $y \in Y$ are incomparable. Then, there is a vertex xy in the graph G . If the vertex xy is labelled 0, then we assign $x < y$; otherwise, the vertex is labelled 1, and we assign $y < x$. Given any linear extension of the order, there is a labelling which corresponds to this

extension. The labelling which corresponds to the linear extension L is the one in which, xy is labelled 0 if and only if $x < y$ in L . The labellings corresponding to different linear extensions are distinct, since any pair of distinct extensions differ in their ordering of at least one incomparable pair of elements. Starting with a valid labelling of the graph, the corresponding linear extension is easily constructed.

Suppose that $xy, x'y \in G$, with $x < x'$. Then, xy and $x'y$ are adjacent vertices in G , and there is a horizontal edge from xy to $x'y$. There are four possible labellings of this edge, namely, $0-0$, $0-1$, $1-0$ and $1-1$. (The labelling of an edge refers to the labelling of the endpoints of the edge, from left to right for horizontal edges, and from bottom to top for vertical edges.) Three of the labellings of this edge are valid, the remaining one, $1-0$, is not. The labelling $1-0$ implies that $y < x$ and $x' < y$ in a linear extension, which is not possible, as $x < x'$. Therefore, no horizontal edge may be labelled $1-0$. A similar argument shows that no vertical edge may be labelled $0-1$. In fact, these are the only restrictions on a labelling. We show that a labelling of G with no horizontal edges labelled $1-0$, and no vertical edges labelled $0-1$, corresponds to a linear extension of P .

The preceding discussion is very reminiscent of the previous chapter. The vertices of the graph are the incomparable pairs of the order. Labelling the vertices of the graph assigns comparisons to the incomparable pairs of the order. An edge of the graph, along with its endpoints, corresponds to a three-element suborder of the order. That suborder is $2+1$, one of the three-element critical orders. The main differences are: only those three-element critical orders which are cover-preserving subsets of the order are being

considered; for width two orders it is possible to efficiently decide which labellings are valid, and to compute the exact number of linear extensions.

Proposition 4.2 *Let G be the graph of the order P . Any 0,1 labelling of the vertices of the graph such that no horizontal edge is labelled 1-0, and no vertical edge is labelled 0-1, corresponds to a linear extension of P .*

Proof Suppose that a labelling of the graph satisfies the given conditions. This labelling defines the relation between any pair of incomparable elements. It must be shown that this relation is transitive. Let $y \in Y$. Since there are no horizontal edges labelled 1-0, the vertices in row y form a sequence of 0's followed by a sequence of 1's, either of which may be empty. (If there are no vertices in row y , then y is a splitting element. In this case, y is already comparable to all other elements of the order. Since there are no vertices involving y , the labelling does not assign any comparisons involving y .) By the definition of the labelling, y is placed above all x , with xy labelled 0, and below all x with xy labelled 1. Therefore, there exists k , such that $y > x_i$ for all $i \leq k$, and $y < x_i$ for all $i > k$. The labelling places y between x_k and x_{k+1} . (Above x_k , or below x_{k+1} , if $k = p$ or $k = 0$ respectively.) Therefore, this labelling merges the element y with the chain X in a valid way. This is true for each $y \in Y$.

These mergings are shown to be consistent with each other. That is, they do not attempt to rearrange the elements of Y . For example, if the labelling places y_j above x_k , then it cannot place y_{j+1} below x_k . For $x \in X$, an argument similar to the previous one, but applied to column x , shows that each x is merged with Y in a valid way. Therefore, when the elements

of Y are merged with X , they are not rearranged. Thus, the labelling represents a valid merging of X and Y , and hence corresponds to a linear extension of P . \square

Thus, a valid 0,1 labelling of G is one in which each row in the graph is a sequence of 0's followed by a sequence of 1's, such that the leading 1 in each row is above, or to the right of, the leading 1 in the previous row. It is possible for either of these sequences to be empty. A similar condition applies to the columns of the graph.

The number of linear extensions of the order is determined by enumerating the valid labellings. The calculation considers the position of the last 0 in each row of the graph. A value $c(i, j)$, is assigned to each vertex. The value of vertex x_j is the number of labellings of the rows, up to and including row y , for which the last 0 in row y occurs at or to the left of x_j . It is simpler to label, and assign values to, all points in the grid, not just those at which there are vertices. However, two labellings are distinct if and only if they differ at some point at which there is a vertex. (Given a valid labelling, there is an equivalent labelling in which the labels of points at which there is no vertex agrees with the ordering of the elements in G .)

To simplify development of the calculation, a new first row and column are added to the graph. In the order, this corresponds to adding a pair of elements x_0 and y_0 , below all other elements. The elements x_0 and y_0 are necessarily comparable; otherwise the number of linear extensions would change. However, the actual comparison between x_0 and y_0 is unimportant, and is never specified. Algorithmically, adding this pair of elements corresponds to initialization. Since x_0 is less than everything else (except,

possibly y_0), all points in the first column are labelled 0. This forces each row to contain at least one 0. Similar reasoning shows that all points in the first row are labelled 1. This allows all valid labellings of the graph, for if a point in the first row is labelled 0, then no vertex in that column can be labelled 1. (The label of the point x_0y_0 is not specified.) Therefore, the value of each point in the first row and column is 1.

There are three possible cases when calculating the value at a point in the graph. Suppose that xy is the current point, at (i, j) , with $u \prec x$ in X and $v \prec y$ in Y .

Case (i) x is incomparable to y .

There is a vertex at xy . The number of ways that the last 0 in row y occurs at or before xy is the number of ways that it occurs before xy , the value at uy , plus the number of ways that it occurs at xy , the value at xv . Therefore, $c(i, j) = c(i - 1, j) + c(i, j - 1)$.

Case (ii) $x < y$.

There is no vertex at xy . Additionally, there are no vertices above and to the left of xy . Therefore, all points above and to the left of xy which are labelled 1 can be relabelled 0, yielding an equivalent labelling. Thus, any labelling with the last 0 in row y occurring before xy is equivalent to one with the last 0 at xy . However, there may be vertices below xy . If such a vertex is labelled 0, then xy cannot be labelled 1. Therefore, there are labellings in which the last 0 may be at xy but not before it. Thus, the value at xy is the number of ways that the last 0 in row y can occur at xy , which is the value at xv . That is, $c(i, j) = c(i, j - 1)$.

Case (iii) $x > y$.

There is no vertex at xy . Additionally, there are no vertices below and to the right of xy . Reasoning similar to that for *Case (ii)* shows that the value at xy is the number of ways in which the last 0 in row y occurs before xy . This is the value at uy , so $c(i, j) = c(i - 1, j)$.

The previous discussion proves the following,

Theorem 4.2 (Width Two) $e(P) = c(p, q)$.

In fact, constructing the graph is unnecessary, as the required information, namely, the relations between pairs of elements, is contained in the order. The calculation was developed by considering the rows of the graph; however, it is easily seen that the calculation is symmetric with respect to the rows and columns of the graph. To calculate the value at a vertex xy , it is sufficient to know the values of the vertex immediately below xy , and the vertex immediately to the left of xy .

The calculation immediately yields the following simple algorithm for calculating the number of linear extensions of a width two order.

```

for all  $i$  do  $c(i, 0) = 0$ 
for all  $j$  do  $c(0, j) = 0$ 
for  $y_j = y_1$  to  $y_q$  do
  while  $x_i < y_j$  do  $c(i, j) = c(i - 1, j)$ 
  while  $x_i || y_j$  do  $c(i, j) = c(i - 1, j) + c(i, j - 1)$ 
  while  $x_i > y_j$  do  $c(i, j) = c(i, j - 1)$ 

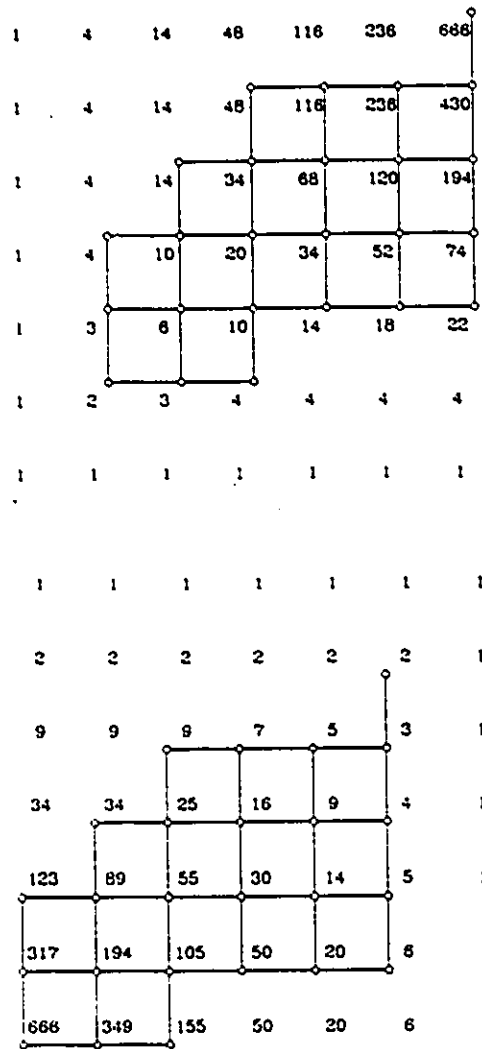
```

It is immediate that this algorithm requires $O(pq)$ time and space to run. Since $p + q = n$, the complexity of this algorithm is $O(n^2)$. The algorithm, as presented, assumes that a chain decomposition is supplied. This is not a problem as a chain decomposition can be found in $O(n^2)$ time as well. The space required by the algorithm can be reduced to $O(n)$, but with the loss of intermediate results.

Our algorithm is similar to that of [Atkinson/Chang 87] (c.f. Theorem 1.14). The main difference is that their algorithm, in effect, counts the number of labellings in which the last 0 in a row occurs at the vertex xy . Our algorithm counts the labellings with the last 0 either at or before the vertex xy . Doing the calculation our way furnishes extra information about the order, by way of the intermediate results. The Atkinson/Chang algorithm is derived by considering mergings of the chains of the order. Ours considers this indirectly, through the graph. The graph gives us more insight into the properties of the order, as is shown in the next chapter.

Rotating the graph G of P by 180 degrees yields the graph of the dual of P . Since P and its dual have the same number of linear extensions, the calculation performed on the graph of the dual yields the same result. However, the intermediate results are different, and having both sets of intermediate results provides more detailed information about the order. The calculation for the dual may be performed directly on the graph G . Instead of starting in the lower left corner of the graph, we start in the upper right one, and modify the rules for calculating the value at a vertex accordingly. (See Figure 4.2) Let $c'(i, j)$ denote the value at point (i, j) in the dual calculation.

82 CHAPTER 4. LINEAR EXTENSIONS OF WIDTH TWO ORDERS



Calculation of the number of linear extensions for the ordered set of Figure 4.1 and its dual

Figure 4.2: Linear extension calculation

Recall that, for $x \in X$ and $y \in Y$,

$$\mathit{init}(xy) = \{u \in X : u \leq x\} \cup \{v \in Y : v \leq y\},$$

$$\mathit{final}(xy) = \{u \in X : u \geq x\} \cup \{v \in Y : v \geq y\},$$

both with the induced ordering.

Proposition 4.3 *Let c and c' be the calculation matrices for P and its dual, respectively. Then*

$$\begin{aligned} e(\mathit{init}(x_i y_j)) &= c(i, j), \\ e(\mathit{final}(x_i y_j)) &= c'(i, j). \end{aligned}$$

Proof Consider the subgraph of G on the set of all points (a, b) such that $a \leq i$ and $b \leq j$. This is the graph for the order $\mathit{init}(x_i y_j)$. Since $c(i, j)$ depends only upon those $c(a, b)$ for which $a \leq i$ and $b \leq j$, clearly $e(\mathit{init}(x_i y_j)) = c(i, j)$. Since the final set $\mathit{final}(x_i y_j)$ in P , is the initial set $\mathit{init}(x_i, y_j)$ in the dual of P , the second result follows from the first. \square

It is interesting to consider how the algorithm works when P is decomposable. Suppose that $x_i y_j$ is the lower left vertex of a component of G , other than the first component, i.e. $(i, j) \neq (1, 1)$. Let e be the value of the upper left vertex in the previous component. This is the number of linear extensions of $\mathit{init}(x_{i-1} y_{j-1})$. Consider the subgraph consisting of the points $x_k y_l$, for $k \geq i-1, l \geq j-1$. This is the graph of $\mathit{final}(x_i y_j)$, with row y_{l-1} as the added first row, and column x_{k-1} as the added first column. It is easily verified that, for $k \geq i-1$ and $l \geq j-1$; $c(i-1, l) = c(k, j-1) = e$. That is, all points in the first row and column of the graph have value e , instead of 1. (See Figure 4.3.) The effect of this is to multiply the number

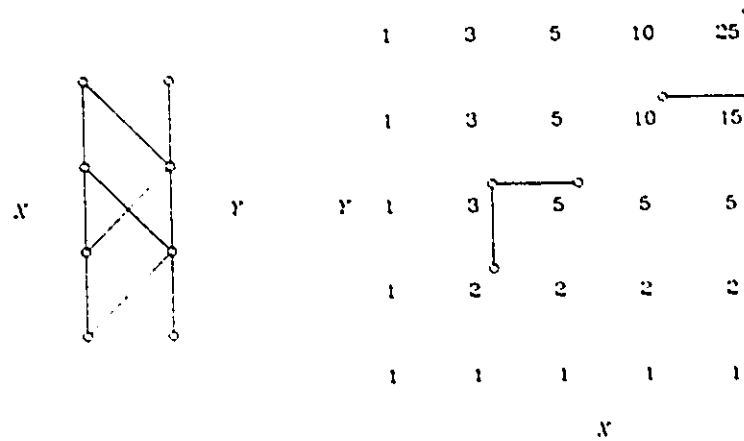


Figure 4.3: Extension calculation for a decomposable order

of linear extensions of $final(x, y)$ by e , which is how the number of linear extensions of a decomposable order is calculated.

4.3 Width k Orders

The algorithm for calculating the number of linear extensions of width two orders may be reinterpreted in terms of initial sets. Instead of counting labellings of the graph of the order — we don't even consider graph — we compute the number of linear extensions of the initial sets of the order. This yields an immediate generalization of the algorithm to greater width orders. If P is a width k order with chain decomposition X_1, X_2, \dots, X_k , then the algorithm computes the number of linear extensions of P using $O(\prod_{i=1}^k |X_i|)$ time and space. Thus,

Theorem 4.3 (Width k) *The number of linear extensions of an n -element width k order can be calculated in time $O(n^k)$.*

It is not clear how our graphical representation of width two orders could be extended to orders with width greater than two. For a width k order, we would attempt to construct the graph on the "grid" formed by the product of k chains. Points in such a grid represent k -element sets, rather than two-element sets. In the graph of a width two order, there is a vertex at a point if and only if the corresponding pair of elements are incomparable. However, for width k orders, it is not simply a question of whether a pair of elements are incomparable, but which pairs of elements in a k -element set are incomparable. Representing all pairs of elements in this k -element set would require $\binom{k}{2}$ different types of vertices. Fortunately, the interpretation of the algorithm for width 2 orders, in terms of initial sets, does not require the graph, and may be extended to greater width orders.

The algorithm for width two orders may be viewed as a dynamic programming approach, applying the formula

$$e(P) = \sum_{x \in \max(P)} e(P \setminus x),$$

to compute the number of linear extensions of the initial sets of the order. In the sum, the term $e(P \setminus x)$ is the number of linear extensions of P in which x is the top element of the linear extension. Summing over all maximals counts the total number of linear extensions. The key point, when considering the initial sets, is that deleting a maximal element of an initial set yields another initial set. Deleting x_i or y_j from the initial set $init(x_i y_j)$ yields either the initial set $init(x_{i-1} y_j)$ or $init(x_i y_{j-1})$, respectively. To

decide if x_i is a maximal element of this initial set, we need only test if it is less than y_j . Similarly for y_j . Thus, the number of linear extensions of the initial set $init(x_i, y_j)$ is calculated by:

- if $x_i < y_j$ then $e(init(x_i, y_j)) = e(init(x_i, y_{j-1}))$;
- if $y_j < x_i$ then $e(init(x_i, y_j)) = e(init(x_{i-1}, y_j))$;
- if $x_i \parallel y_j$ then $e(init(x_i, y_j)) = e(init(x_i, y_{j-1})) + e(init(x_{i-1}, y_j))$.

The algorithm simply computes the number of linear extensions of $init(x_i, y_j)$, by computing the number of extensions in which each of x_i and y_j is the top element. This is more apparent when the algorithm is restated as follows. Initially $c(i, j) = 0$,

- if $x_i \not\prec y_j$ then $c(i, j) = c(i, j) + c(i-1, j)$ — this counts the linear extensions of $init(x_i, y_j)$ in which x_i is the top element;
- if $y_j \not\prec x_i$ then $c(i, j) = c(i, j) + c(i, j-1)$ — this counts the linear extensions of $init(x_i, y_j)$ in which y_j is the top element.

This is trivially generalizable to orders with width greater than two. Let P be a width k order with chain decomposition X_1, X_2, \dots, X_k , where $X_i = x_{i,1} < x_{i,2} < \dots < x_{i,h_i}$. The generalization of the calculation is as follows. Let $I = init\{x_{1,j_1}, x_{2,j_2}, \dots, x_{k,j_k}\}$ be an initial set in P . If $x_{l,j_l} \not\prec x_{i,j_i}$, for any i , then there are linear extensions of this initial set in which x_{l,j_l} is the top element. The number of such extensions is

$$e(I \setminus x_{l,j_l}) = e(init\{x_{1,j_1}, x_{2,j_2}, \dots, x_{l-1,j_{l-1}}, x_{l,j_l-1}, x_{l+1,j_{l+1}}, \dots, x_{k,j_k}\}).$$

Summing this value over all l such that $x_{l,j_l} \not\prec x_{i,j_i}$, for any i , gives the number of linear extensions of I . For convenience, the set of elements

$\{x_{1,0}, x_{2,0}, \dots, x_{k,0}\}$ is added below everything else in the order. As in the width two case, these elements are linearly ordered, but the ordering is unimportant. The calculation is done using a k -dimensional array c , with the i th index ranging from 0 to h_i .

The algorithm for width k orders is,

```

for  $j_1, j_2, \dots, j_i, \dots, j_k = 0$  to  $h_i$  do  $c(j_1, j_2, \dots, j_k) = 0$ 
for  $j_1, j_2, \dots, j_i, \dots, j_k = 0$  to  $h_i$  do
  for  $l = 1$  to  $k$  do
    if  $x_{l,j_l} \neq x_{i,j_l}$  (for all  $i$ ) then
       $c(j_1, j_2, \dots, j_k) = c(j_1, j_2, \dots, j_k) + c(j_1, j_2, \dots, j_l - 1, \dots, j_k)$ 

```

The complexity of this algorithm is clearly $O(\prod_{i=1}^k |X_i|) = O(n^k)$. This is an improvement of the $O(n^{k+1})$ algorithm in [Steiner 90]. The improvement in performance results from the simplicity of initial sets as opposed to ideals.

The statement of the algorithm assumes that a chain decomposition is given. This does not change the complexity, since a chain decomposition may be found in time $O(n^2)$.

The calculation for a width three order is demonstrated in Figure 4.4.

88 CHAPTER 4. LINEAR EXTENSIONS OF WIDTH TWO ORDERS

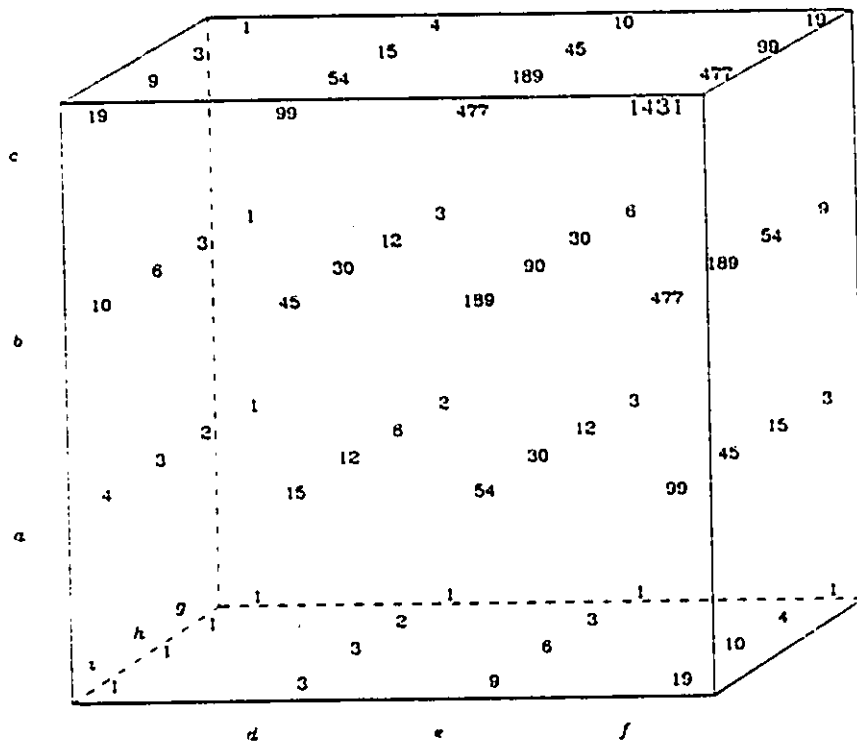
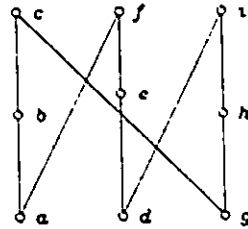


Figure 4.4: Extension calculation for width three order

Chapter 5

Unimodality and Probability

In the previous chapter the number of linear extensions of a width two order was calculated by enumerating labellings of a graph drawn on a grid. The algorithm which performs this calculation is similar to the algorithm of [Atkinson/Chang 87]. While the implementation of the two algorithms is somewhat similar, the development is quite different. The benefit of our method is twofold. The graph provides information about the order, which is used to prove theoretical results about width two orders. In addition, the intermediate results of the calculation give structural information about the linear extensions of the order.

The graph is used to show that sequences of the number of linear extensions of certain suborders of a width two order are unimodal. Once the number of linear extensions of a width two order and its dual have been calculated, the probabilities of certain comparabilities or covering relations occurring in a linear extension are efficiently calculated. This is

very useful for the well-known problem of finding a pair of elements x, y for which $\text{prob}(x < y)$ is close to $1/2$. A related problem is that of proving that $\delta(P) \geq 1/3$ for any order P ; that is, in any order P , there exists a pair of elements x, y satisfying $1/3 \leq \text{prob}(x < y) \leq 2/3$. This was proved, for width two orders, in [Linial 84] (Theorem 1.11), and those width two orders for which $\delta(P) = 1/3$ were characterized in [Aigner 85] (Theorem 1.12). A shorter proof of this result is possible, by means of our graphical representation of width two orders.

Recall that c and c' are the matrices for the calculations of the number of linear extensions of P and its dual, respectively, as defined in the preceding chapter. (See pages 78 and 81.)

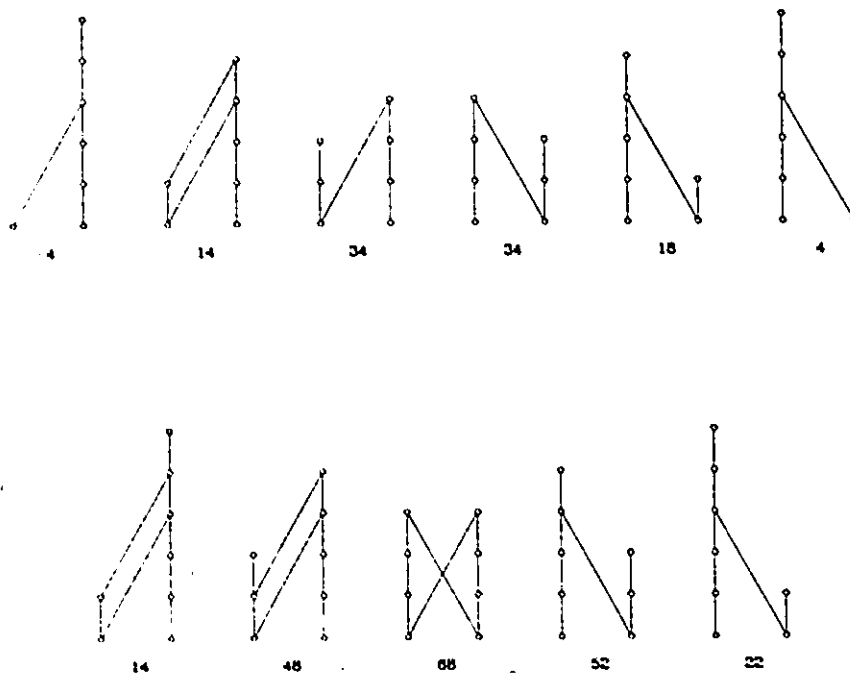
5.1 Unimodal Sequences

Suppose that $P = X + Y$ is a disjoint sum of the chains $X = x_1 < x_2 < \dots < x_p$ and $Y = y_1 < y_2 < \dots < y_q$. Choose k , ($1 < k < p + q$) and consider the sequence

$$e(\text{init}(x_0 y_k)), e(\text{init}(x_1 y_{k-1})), e(\text{init}(x_2 y_{k-2})), \dots, e(\text{init}(x_k y_0)).$$

(If $k > p$ or $k > q$, then some of the initial or final terms of this sequence are not defined and are not included.) The orders in this sequence are the k -element initial sets of P , ordered by the size of the subchain from X . Each successive order in the sequence is found by removing the top element from the subchain of Y , and adding one more element to the top of the subchain of X . Since

$$e(\text{init}(x_i y_{k-i})) = \binom{k}{i},$$



Unimodal sequences for $k = 7, 8$, for the width two order of Figure 4.1

Figure 5.1: Unimodal sequences

the sequence is a unimodal sequence. Moreover, this sequence is strictly increasing, then strictly decreasing, although, the final term of the increasing subsequence may equal the initial term of the decreasing one.

In fact, a similar unimodality property holds for any width two ordered set. For the special case of indecomposable orders, the result is exactly the same as for a disjoint sum of chains. (See Figure 5.1.)

Theorem 5.1 (Unimodal Sequence) *For an indecomposable width two*

ordered set P , and $1 < k < |P|$, the sequence

$$c(\text{init}(x_0 y_k)), c(\text{init}(x_1 y_{k-1})), c(\text{init}(x_2 y_{k-2})), \dots, c(\text{init}(x_k y_0)),$$

is a unimodal sequence which is strictly increasing, then strictly decreasing.

Proof (Unimodal Sequence) If both the increasing and decreasing sequences have strict inequalities, then the maximum value of the sequence may occur (at most) twice. In fact, while the maximum value may occur twice, this cannot happen for consecutive values of k . In addition, the suborders at which the maximum values occur have a pair of maximal elements, not a top element.

The sequence in question is a diagonal in the matrix c , running down to the right, so that the sum of the indices is k . The result is proved by induction on k . Since P is indecomposable, it has a pair of minimal elements. Therefore, if $k = 2$ the sequence is 1,2,1. This sequence satisfies all the conditions of the theorem.

The result is proved for $k > 2$, assuming that it has been proved for all smaller values. Consider the k th diagonal of c . We show that: the maximum value occurs at a vertex; it can occur at most twice; it occurs only once if the maximum value in the previous diagonal was repeated. It is then shown that traversing the diagonal, away from the maximum, in either direction, yields a strictly decreasing sequence. This proves the theorem.

By the row increasing property, there are no gaps between the vertices in any diagonal. The values at the vertices are sums of pairs of consecutive terms of the unimodal sequence in the previous diagonal. We will show that such a sequence must itself be unimodal. Furthermore, the maximal

value in such a unimodal sequence must result from a sum which includes the maximal value from the previous sequence.

Suppose that the unique maximum m , in the previous diagonal, occurred at the vertex (i, j) . Since G is row increasing and connected, there must be a vertex at one, or both, of the points $(i, j + 1)$ or $(i + 1, j)$, in the current diagonal. The value at such a vertex is a sum which includes m . Such a value is clearly maximal, since no other sum includes m . If there are vertices at both of these points, and $c(i - 1, j + 1) = c(i + 1, j - 1)$ (in the previous diagonal), then the maximum occurs twice. If there is a vertex at only one point, then its value is the unique maximum.

Now, suppose that the maximum m , in the previous diagonal, occurred twice, at the vertices (i, j) and $(i + 1, j - 1)$. As G is connected, there is a vertex at $(i + 1, j)$, whose value is $2m$. This is the unique maximum in the current diagonal. Since the maxima always occur at vertices, the corresponding suborders have a pair of maximals, not a top element.

Now, we show that, moving outward from the maximum in either direction, yields a strictly decreasing sequence. This proves that the diagonal is a unimodal sequence. By symmetry, this need only be shown for one direction, say for the subsequence going down to the right. Suppose that the maximum in the current diagonal is at (i, j) , with the second maximum (if present) at $(i - 1, j + 1)$. Thus, $c(i, j) > c(i + 1, j - 1)$, since they cannot be equal. We now show that, for $l > 1$

$$c(i + l, j - l) > c(i + l + 1, j - l - 1)$$

Let $c_1 = c(i + l - 1, j - l)$, $c_2 = c(i + l, j - l - 1)$ and $c_3 = c(i + l + 1, j - l - 2)$, from the previous diagonal. Since the maximum in the previous diagonal

occurred at $(i-1, j)$ or $(i, j-1)$, these three values form a strictly decreasing sequence. Now, $c(i+l, j-l) = c_1$ or $c_1 + c_2$, and $c(i+l+1, j-l-1) = c_2$ or $c_2 + c_3$. However, $c(i+l+1, j-l-1)$ is a sum only if $c(i+l, j-l)$ is also a sum. Therefore, we conclude that $c(i+l, j-l) > c(i+l+1, j-l-1)$. \square

If P is decomposable, the sequence is still unimodal, although it may not be strictly increasing or strictly decreasing.

5.2 Calculating Probabilities

The calculation algorithm yields the number of linear extensions of an ordered set. To calculate the probability that a certain property holds in an arbitrary linear extension, it is only necessary to calculate the number of linear extensions satisfying that property.

Theorem 5.2 (Probabilities) *Let c, c' be the calculation matrices of P and its dual. Then, for x_i and y_j , incomparable in P :*

- (i) $e(x_i < y_j) \geq c(i, j-1)c'(i+1, j)$ with equality if and only if x_{i+1} and y_{j-1} are comparable;
- (ii) $e(y_j < x_i) \geq c(i-1, j)c'(i, j+1)$ with equality if and only if x_{i-1} and y_{j+1} are comparable;
- (iii) $e(x_i < y_j) = e(y_j < x_i) = c(i-1, j-1)c'(i+1, j+1)$;
- (iv) $e(x_{i-1} < y_j < x_i) = c(i-1, j-1)c'(i, j+1)$;
- (v) $e(y_{j-1} < x_i < y_j) = c(i-1, j-1)c'(i+1, j)$.

Cases (i) and (ii) assume that the extra (splitting) elements x_0, y_0, x_{p+1} and y_{q+1} have been added to the order.

Proof (Probabilities) The proofs are all based on finding the numbers of linear extensions of an initial set below the given elements, and a final set above the elements. Then, linear extensions of P are found by merging the linear extensions of these initial and final sets. The key point is the fact that the number of linear extensions of all initial and final sets are contained in the matrices c and c' , respectively. Indeed, $e(\text{init}(x_i y_j)) = c(i, j)$ and $e(\text{final}(x_i y_j)) = c'(i, j)$.

Let L_1 and L_2 be linear extensions of $\text{init}(x_i y_{j-1})$ and $\text{final}(x_{i+1} y_j)$, respectively. Then, $L_1 * L_2$ is a linear extension of P in which $x_i < y_j$. Therefore, $c(i, j-1)c'(i+1, j) \leq e(x_i < y_j)$. However, equality occurs if and only if there are no mergings of L_1 and L_2 , with $x_i < y_j$. In such a merging, $x_{i+1} < y_{j-1}$ must occur. Therefore, equality holds if and only if $x_{i+1} > y_{j-1}$. (Since $x_i \parallel y_j$ implies that $x_{i+1} \not< y_{j-1}$, in P .) This proves (i); the proof of (ii) is similar.

Suppose that L is an extension of P in which $x_i < y_j$ or $y_j < x_i$. In L , below x_i and y_j , is a linear extension L_1 , of $\text{init}(x_{i-1} y_{j-1})$; above them is a linear extension L_2 of $\text{final}(x_{i+1} y_{j+1})$. These extensions are separated by x_i and y_j . Thus, L is of the form $L_1 * \{x_i, y_j\} * L_2$ (with the appropriate comparison between x_i and y_j). Therefore, $e(x_i < y_j) = e(y_j < x_i) = c(i-1, j-1)c'(i+1, j+1)$, proving (iii).

Let L_1 be an extension of $\text{init}(x_{i-1} y_{j-1})$, and L_2 be an extension of $\text{final}(x_i y_{j+1})$. Then, $L_1 * \{y_j\} * L_2$ is a linear extension of P in which $x_{i-1} < y_j < x_i$. Moreover, every linear extension of P with $x_{i-1} < y_j < x_i$ is of this form. Therefore, $e(x_{i-1} < y_j < x_i) = c(i-1, j-1)c'(i, j+1)$. The proof of (v) is similar. \square

Consider the problem of calculating probabilities when equality does not hold in cases (i) and (ii) of Theorem 5.2. Since $e(P) = e(x < y) + e(y < x)$, the only case when $e(x_i < y_j)$ cannot be calculated exactly, is when there are vertices at both $(i+1, j-1)$ and $(i-1, j+1)$. In this case, the row increasing property implies that $x_i y_j$ is in the center of a 3×3 square of vertices. This means that P contains two pairwise incomparable three-element chains, with x_i and y_j being the central elements of the chains. When calculating $e(x_i < y_j)$, only those extensions with $x_{i+1} < y_{j-1}$ are not counted. Therefore,

$$\begin{aligned} e(x_i < y_j) &= c(i, j-1)c'(i+1, j) + e(x_{i+1} < y_{j-1}) \\ &\geq c(i, j-1)c'(i+1, j) + c(i+1, j-2)c'(i+2, j-1). \end{aligned}$$

If $e(x_{i+1} < y_{j-1})$ cannot be calculated exactly, it must be because there is a vertex at $(i+2, j-2)$. Similarly, if $e(y_j < x_i) = c(i-1, j)c'(i, j+1) + e(y_{j+1} < x_{i-1})$ cannot be calculated exactly, then there is a vertex at $(i-2, j+2)$. Thus, if it is not possible to calculate $e(x_i < y_j)$ in this way, then $x_i y_j$ is in the center of a 5×5 square of vertices. That is, P contains two mutually incomparable five-element chains, with x and y as the central elements. This is already a very particular requirement on the structure of P . In any case, it is always possible to calculate $e(x_i < y_j)$ exactly by taking a sum of terms of the form $e(y_{l-1} < x_i < y_l)$ or $e(x_{l-1} < y_j < x_l)$.

Theorem 5.2 states that, if x_i and y_j are incomparable in P , then $e(x_i < y_j) = e(y_j < x_i) = c(i-1, j-1)c'(i+1, j+1)$. The same argument shows that, if $x_i < y_j$ in P , then $e(x_i < y_j) = c(i-1, j-1)c'(i+1, j+1)$. (Note that even though $x_i < y_j$ in P , that need not be the case in all linear extensions.) When $x_i < y_j$, it is possible to delete the comparability between x_i and

y_j , without affecting any other comparisons. Let $Q = P \setminus (x_i < y_j)$ be the order in which this comparability has been deleted. Thus, P is a partial extension of Q . If $y_j < x_i$ in a linear extension of Q , then $y_j < x_i$ in that linear extension. The reason for this is that the only elements which could possibly separate x_i and y_j are x_{i-1} and y_{j+1} , if they exist. However, if x_{i-1} exists, then $x_{i-1} < y_j$, since no other relations were changed. Similarly, if y_{j+1} exists, then $y_{j+1} > x_i$. Therefore, $e(Q) = e(P) + e(Q; y_j < x_i) = e(P) + c(i-1, j-1)c(i+1, j+1)$. That is, when $x_i < y_j$ in P , the number of linear extensions of the order which results from making x_i and y_j incomparable, is easily calculated.

The results of the theorem are also useful for the problem of balancing the linear extensions of an order. The objective is to find a pair of elements x, y , for which $\text{prob}(x < y)$ is close to $1/2$. Ideally, we would like to find a pair of elements which realizes $\delta(P) = \max\{\text{prob}(x < y) : \text{prob}(x < y) \leq 1/2\}$. Linial (Theorem 1.11) proved that, for width two orders there is always a pair x, y for which $1/3 \leq \text{prob}(x < y) \leq 2/3$. In fact, his proof shows more than what his theorem states. Essentially, he proves that, if $\text{prob}(x_1 < y_1) < 1/3$, then there exists y with $1/3 < \text{prob}(x_1 < y) < 2/3$. Therefore, if $\text{prob}(x_1 < y_1) \neq 1/3, 2/3$, then $\delta(P) > 1/3$. Also, there is always a pair x, y with $1/3 \leq \text{prob}(x, y) \leq 2/3$, in which one element is minimal. Moreover, $\text{prob}(x_1 < y_1)$ determines if such a pair includes x_1 or y_1 .

The main outstanding problem is that of efficiently finding a pair x, y , for which $1/3 \leq \text{prob}(x < y) \leq 2/3$, or finding a pair which realizes $\delta(P)$. Linial describes an algorithm similar to the following naïve algorithm. Use

our algorithm to calculate $e(P)$. Now, for each pair x, y add the comparability $x < y$ to P (and take the transitive closure); then use the algorithm to calculate $e(x < y)$. This is inefficient since each query of the probability for a pair of elements takes time pq , the running time of the algorithm for counting linear extensions. This can be improved significantly by taking advantage of the extra information our algorithm provides.

After an initial $O(pq)$ calculation to determine the number of linear extensions of the order, many queries can be answered in constant time. In particular, the queries can always be answered in constant time when one of x_i or y_j is maximal or minimal. Therefore, it is always possible to find a pair of elements for which the probability is between $1/3$ and $2/3$, in time $O(pq)$. If $\text{prob}(x_i < y_j)$ cannot be calculated in constant time, it can be calculated in time $1/2 \min\{p, q\}$ by taking an appropriate sum of terms of the form $\text{prob}(y_{i-1} < x_i < y_i)$ or $\text{prob}(x_{i-1} < y_j < x_i)$. Therefore, an optimal pair, one which realizes $\delta(P)$, can always be found in time $O(pq \min\{p, q\})$.

When $\text{prob}(x_i < y_j)$ cannot be calculated in constant time, it can be approximated. Let m and M be the maximum and minimum respectively of $c(i, j-1)c'(i+1, j)$ and $c(i-1, j)c'(i, j+1)$. Since both m and M must increase, the worst case is when m increases by only 1 and M increases to $e(P) - m - 1$. Therefore, $m+1 \leq e(x_i < y_j) \leq e(P) - m - 1$, or equivalently

$$\frac{m+1}{e(P)} \leq \text{prob}(x_i < y_j) \leq 1 - \frac{m+1}{e(P)}.$$

5.3 Extremal Width Two Orders

[Linial 84] proved that, for any ordered set of width two, there is always a pair x, y for which $1/3 \leq \text{prob}(x < y) \leq 2/3$.



Figure 5.2: Extremal width two orders for $\delta(P) = \frac{1}{3}$

It is easily seen that, if a width two order P is a linear sum of orders each isomorphic either to 1 or $2 + 1$ (see Figure 5.2), then $\delta(P) = 1/3$. [Aigner 85] characterizes those width two orders P for which $\delta(P) = 1/3$, by proving that the converse is also true (Theorem 1.12). Using our theorems (5.2, 5.1) and the stronger statement of Linial's result, we give a short clear proof of Aigner's result (Theorem 1.12).

Theorem 5.3 (Extremal Width Two Orders) *If P is an ordered set of width two with $\delta(P) = 1/3$, then P must be a linear sum of ordered sets each isomorphic to a singleton or $2 + 1$.*

Proof (Extremal Width Two Orders) Let P be a width two order with $\delta(P) = 1/3$. We may assume that P is indecomposable, since probabilities are unaffected by linear sums. Therefore, the graph of P is connected. We prove that $P \cong 2 + 1$.

The proof considers only some initial part of the graph G of P . This is used to calculate the probabilities of certain comparabilities between pairs of elements of P . All necessary values of the matrix c are readily calculated. For the matrix c' , variables are assigned to represent some values, then the remaining values are expressed in terms of these variables. The notation is simplified by assuming that all values in c' have been divided by $e(P)$.

The result of this is that the calculations yield probabilities, not numbers of linear extensions. The proof considers a number of special cases.

Since P is indecomposable, it has a pair of minimal elements x_1 and y_1 . It is not possible to have $1/3 < \text{prob}(x_1 < y_1) < 2/3$, as this would immediately imply that $\delta(P) > 1/3$. If $\text{prob}(x_1 < y_1) < 1/3$ or $\text{prob}(x_1 < y_1) > 2/3$, then the stronger statement of Linial's theorem gives $\delta(P) > 1/3$. Therefore, $\text{prob}(x_1 < y_1) = 1/3$ or $2/3$. If the vertex at $(1,1)$ is the only vertex in G , then $P = \{x_1, y_1\}$ and $\delta(P) = 1/2$. Therefore, there is another vertex in G . As G is connected, this vertex must be at $(1,2)$ or $(2,1)$.

Suppose that there are vertices at both of these points, and that the values of c and c' are as illustrated in Figure 5.3a. Then,

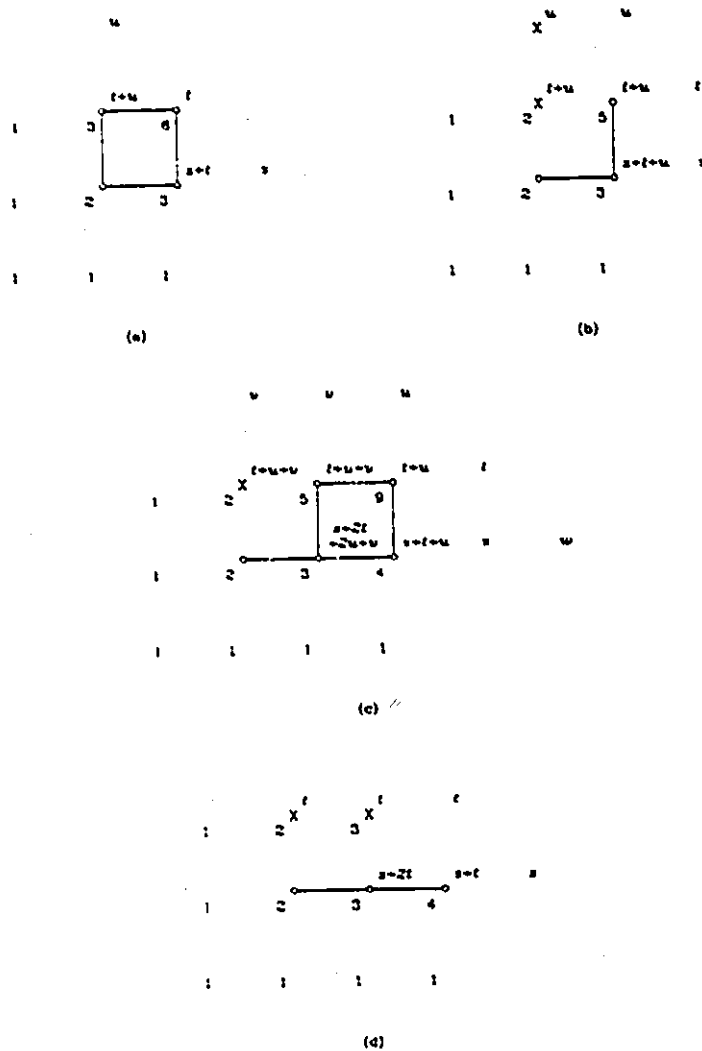
$$\text{prob}(x_1 < y_1) = s + t \quad \text{and} \quad \text{prob}(y_1 < x_1) = t + u.$$

Say $t + u = 1/3$ and $s + t = 2/3$. Therefore, $t < 1/3$ and $1/3 < s < 2/3$. Then $\text{prob}(x_2 < y_1) = s$ implies $\delta(P) > 1/3$, a contradiction. There cannot be vertices both at $(1,2)$ and $(2,1)$.

Assume, without loss of generality, that there is a vertex at $(2,1)$, but no vertex at $(1,2)$. If there are no other vertices in G , then $P \cong 2 + 1$, so assume that there is at least one more vertex. As G is connected, there must be a vertex at $(2,2)$ or $(3,1)$. Suppose that there is a vertex at $(2,2)$. See Figure 5.3b for the graph and the values of c and c' . Then,

$$\text{prob}(x_1 < y_1) = s + t + u \quad \text{and} \quad \text{prob}(y_1 < x_1) = t + u.$$

Clearly $s + t + u = 2/3$ and $t + u = 1/3$, so $t < s = 1/3$. Then $c'(3,1) = s > t = c'(3,2)$ implies $c'(3,1) = c'(3,2) + c'(4,1)$, there is a vertex at $(3,1)$. By the row increasing property, there is also a vertex at $(3,2)$.



Numbers are values of c ; letters are values of c' . The symbol X denotes that there is no vertex at that point.

Figure 5.3: The cases for the proof of Theorem 5.3

Figure 5.3 shows the graph with these new vertices. Now,

$$\text{prob}(x_1 < y_1) = s + 2(t + u) + v \quad \text{and} \quad \text{prob}(y_1 < x_1) = t + u + v.$$

Then, $s + 2(t + u) + v = 2/3$ and $t + u + v = 1/3$, thus $s + t + u = 1/3$ and $s = v$. In addition:

$$\text{prob}(x_2 < y_2) = \text{prob}(x_3 < y_1) + \text{prob}(y_1 < x_3 < y_2) = s + 3t;$$

$$\text{prob}(y_2 < x_3) = \text{prob}(y_2 < x_2) + \text{prob}(x_2 < y_2 < x_3) = 2v + 3u = 2s + 3u.$$

Since $s = c'(4, 1) \geq c'(4, 2) = t$, we conclude $2s + 3u \geq s + t + 3u > s + t + u = 1/3$. Then $\delta(P) = 1/3$ implies $2s + 3u \geq 2/3$, so $s + 3t \leq 1/3$, and $2t \leq u$ (as $s + t + u = 1/3$). Now, $v = c'(2, 3) \geq c'(3, 3) = u$. Therefore, $s = v \geq u > t$, that is $c'(4, 1) > c'(4, 2)$: there is a vertex at $(4, 1)$, so $s = t + w$. But u, t and w are terms in a unimodal sequence (which is strictly increasing and decreasing, by Theorem 5.1), so $u > t$ implies $t > w$. Then $s = t + w < 2t \leq u \leq v$ contradicts $s = v$. Therefore, the assumption that there is a vertex at $(2, 2)$ is incorrect.

The final case to consider is a vertex at $(1, 3)$, but no vertex at $(2, 2)$. Figure 5.3d shows this new situation. In this case,

$$\text{prob}(x_1 < y_1) = s + 2t \quad \text{and} \quad \text{prob}(y_1 < x_1) = t.$$

Trivially, $s + 2t = 2/3$ and $t = 1/3$. This implies that $s = 0$, which is not possible.

Therefore, there cannot be a third vertex and $P \cong 2 + 1$. \square

Chapter 6

Linear Extension Majority Cycles

The height two order of Figure 6.1 contains the linear extension majority cycle $1 > 2 > 3 > 1$. This is proved by a straightforward calculation of the number of linear extensions of the order, using Lemma 6.2. This proves,

Theorem 6.1 (Majority Cycle) *There are height two orders containing linear extension majority cycles.*

It was not previously known if height two orders could contain majority cycles.

The first example of a majority cycle, in [Fishburn 74], was a 31-element height 28 order. A nine-element, height three order with a majority cycle (Figure 6.2a) was presented in [Fishburn 86] and [Ganter/Häfner/Poguntke 87]. In this order, $\text{prob}(1 > 2) = \text{prob}(2 > 3) = \text{prob}(3 > 1) = 80/159 \approx 0.503$. Recently, [Gehrlein/Fishburn 89] showed, via a computer search, that a majority cycles cannot exist in orders with eight or fewer elements. The search showed that there are ex-

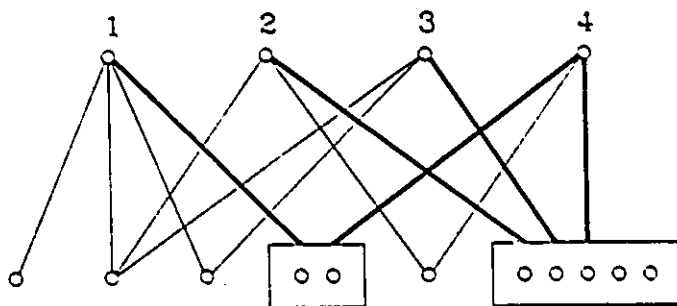
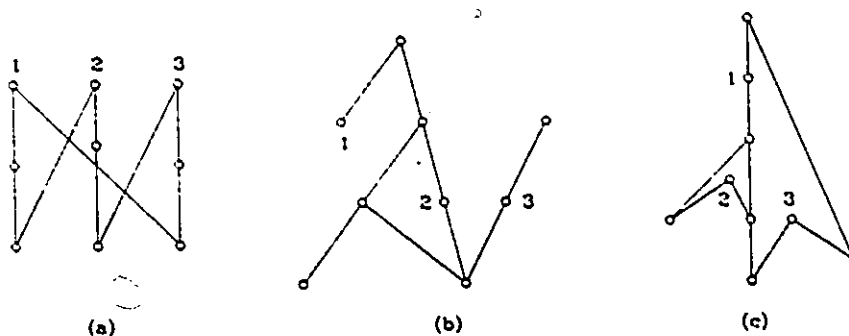
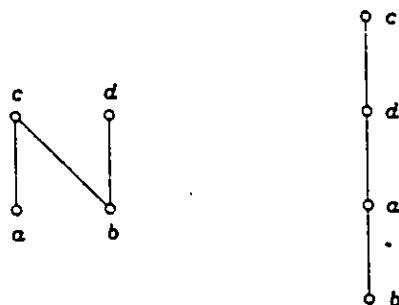
Figure 6.1: Height two order with majority cycle $1 >' 2 >' 3 >' 1$ 

Figure 6.2: Nine-element orders with majority cycles

actly five, nine-element orders with majority cycles, all with height three or more. These orders are illustrated in Figure 6.2.

6.1 Background

Suppose that a set of elements on which there is only a partial ordering must be totally ordered. While any linear extension of the order is acceptable from a theoretical viewpoint, it is unarguable that some extensions may

Figure 6.3: "Natural" linear extension of N

seem more "natural" than others. For example, consider the order N , of Figure 6.3. Of the five possible linear extensions, $b < a < d < c$ is, arguably, the most natural, for several reasons. While $a > d$ is possible in an extension it seems unnatural to place a minimal element above a maximal element; indeed, this occurs in only one linear extension. Since b is covered by both c and d , while a is covered only by c we expect that b should be below a in a linear extension. Similar reasoning applied to c and d says that d should go below c , yielding the indicated linear extension.

The idea being considered is the preference ordering $<'$, which places x below y if and only if x occurs below y in more than half of the linear extensions of the order. The reasoning which gave the comparabilities between a and b , and c and d is supported by the following,

Lemma 6.1 (Preference Order) *Suppose that $x, y \in P$ are incomparable. Define $D'_x = \text{down}(x) \setminus x$ and $U'_x = \text{up}(x) \setminus x$. Define D'_y and U'_y similarly. If $D'_x \subset D'_y$ and $U'_y \subset U'_x$, then $x <' y$. If $D'_x = D'_y$ and $U'_y = U'_x$, then $\{x, y\}$ is an autonomous set.*

The proof is based on the idea of swapping x and y in those extensions in which $x > y$. This is always possible, given the conditions of the lemma. This shows that, for each extension with $x > y$, there is a corresponding extension with $x < y$, so $\text{prob}(x < y) \geq 1/2$. However, if $z \in (D'_y \setminus D'_x) \cup (U'_x \setminus U'_y)$ and $x < z < y$, then x and y cannot be swapped. That is, there are extensions with $x < y$, for which there is no corresponding extension with $x > y$. Therefore, $\text{prob}(x < y) > 1/2$, so $x <' y$. If the conditions of the second statement hold, then x and y may be interchanged in any linear extension, so clearly $\text{prob}(x < y) = 1/2$.

The lemma still applies if x and y are comparable: given the other conditions, the only possibility is $x < y$, in which case $\text{prob}(x < y) = 1$.

6.2 Linear Extensions of Height two Orders

The result requires an efficient method of counting the number of linear extensions of a height two order. One method, which is sufficient for our purposes, is to compute the number of linear extensions for each linear ordering of the maximals (or minimals) of the order. We find a partial extension of the order by taking a linear extension of the maximals, then taking the transitive closure. The resulting order consists of a chain, each element of which covers a (possibly empty) antichain, which is comparable only with the elements of the chain above its upper cover, as illustrated in Figure 6.4. The number of linear extensions of such an order is easily calculated using the following lemma. We perform this calculation for all linear orderings of the maximals to compute the number of linear extensions of the order.

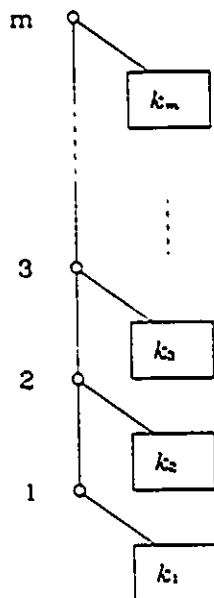


Figure 6.4: Linear extension calculation for height two orders

Lemma 6.2 Let P consist of an m -element chain, $1 < 2 < \dots < m$, with i ($1 \leq i \leq m$) covering a k_i -element antichain which is comparable only with the elements of the chain above i . Let $n = |P| = m + \sum_{i=1}^m k_i$, then

$$e(P) = \frac{(n-1)!}{\prod_{i=1}^{m-1} (k_1 + k_2 + \dots + k_i + i)}$$

Proof (Lemma 6.2) To form a linear extension of P , we find linear extensions of the antichains, then merge the resulting chains, working from the bottom up. The total number of linear extensions of all antichains is $(k_1!)(k_2!) \dots (k_m!)$. Below 2 there are $k_1 + 1$ - and k_2 -element chains. These

may be merged in

$$\binom{k_1 + k_2 + 1}{k_1 + 1}$$

different ways. The result is a $k_1 + k_2 + 2$ -element chain, below 3, which includes 2. This chain may be merged with the k_3 -element chain below 3 in

$$\binom{k_1 + k_2 + k_3 + 2}{k_1 + k_2 + 2}$$

different ways. Continuing this process till we have a linear extension yields

$$\begin{aligned} e(P) &= (k_1!)(k_2!) \cdots (k_m!) \binom{k_1 + k_2 + 1}{k_2 + 1} \binom{k_1 + k_2 + k_3 + 2}{k_1 + k_2 + 2} \cdots \\ &\quad \cdots \binom{k_1 + k_2 + \cdots + k_m + m - 1}{k_1 + k_2 + \cdots + k_{m-1} + m - 1} \\ &= \frac{(n-1)!}{(k_1 + 1)(k_1 + k_2 + 2) \cdots (k_1 + k_2 + \cdots + k_{m-1} + m - 1)}. \end{aligned}$$

□

Proof (Majority Cycle) Applying Lemma 6.2 to the order of Figure 6.1

yields $e(P) = 1\,703\,630\,880$ and

$$\begin{aligned} e(1 > 2) &= 852\,626\,880 \Rightarrow \text{prob}(1 > 2) \approx 0.500476, \\ e(2 > 3) &= 851\,891\,040 \Rightarrow \text{prob}(2 > 3) \approx 0.500044, \\ e(3 > 1) &= 852\,163\,200 \Rightarrow \text{prob}(3 > 1) \approx 0.500204. \end{aligned}$$

Therefore, $1 >' 2 >' 3 >' 1$.

□

6.3 Miscellany

It is not known if there are smaller height two orders containing linear extension majority cycles. However, our example has minimal size for an order with only four maximals.

There are some restrictions on which height two orders may contain majority cycles.

For P with height two, define

$$\begin{aligned} A &= \{x \in P : x > y \text{ for some } y \in P\}, \\ B &= \{y \in P : x > y \text{ for some } x \in P\}, \\ C &= X \setminus (A \cup B). \end{aligned}$$

Thus, $A \cup C$ is the set of maximal elements, $B \cup C$ is the set of minimal elements, and C is the set of isolated elements of P . In Figure 6.1, $|A| = 4$, $|B| = 11$, and $|C| = 0$.

Proposition 6.1 *If a height two order P contains a linear extension majority cycle, then the cycle is contained in A , or B . That is, the elements of the cycle are all contained in A , or are all contained in B .*

Proof It is sufficient to show that if (x, y) is contained in any of $(A \times B)$, $(A \times C)$, $(C \times B)$ or $(C \times C)$ then $x \not\prec' y$. For, suppose that there is a cycle in P which is not contained in A or B . We may assume, without loss of generality, that there is such a cycle of the form $x_1 >' x_2 >' \dots >' x_m >' x_1$, where $x_m \in A$ and $x_1 \notin A$. There exists i such that $x_i \notin A$ and $x_{i+1} \in A$. Then $x_i <' x_{i+1}$, a contradiction. Thus, the cycle must be contained entirely in A or B .

If $(x, y) \in (C \times C)$ with $x \neq y$, then, by symmetry, $\text{prob}(x > y) = 1/2$. We show that if $(x, y) \in (A \times B)$ then $x >' y$. The cases for $(x, y) \in (A \times C)$, $(C \times B)$ are similar.

Suppose that $(x, y) \in (A \times B)$. If x is below y in a linear extension of P , then we may form another linear extension by interchanging x and y . This is possible as x is a maximal element of P and so can be "moved up"

in a linear extension, while minimal y can be "moved down". Thus, the number of linear extensions in which x is above y is at least as large as the number in which x is below y . However, there are linear extensions with x above y for which x and y may not be interchanged. As $x \in A$, there exists $z \in P$ with $z < x$. Since $z < y$ is not possible, there is a linear extension of P in which $y \leq z < x$. Now, interchanging x and y would force $x < z$, a contradiction. Therefore, $e(x > y) > e(x < y)$ and $x >' y$. \square

Proposition 6.2 *If a height two order contains a linear extension majority cycle, contained in A , then $|A| \geq 4$.*

Proof Isolated elements in the order do not affect the presence of $>'$ cycles, so we assume that $C = \emptyset$. Suppose that $A = \{1, 2, 3\}$. Let β_i be the number of elements of B which are below $i \in A$, and β_{ij} the number of elements of B below either i or j in A . Define $e(i > j > k)$ to be the number of linear extensions of P in which $i > j > k$. Then, by Lemma 6.2,

$$e(i > j > k) = \frac{(n-1)!}{(\beta_k + 1)(\beta_{jk} + 2j)}.$$

For convenience, let $b_k = (n-1)!/(\beta_k + 1)$ and $b_{jk} = 1/(\beta_{jk} + 2)$, so $e(i > j > k) = b_k b_{jk}$.

Suppose that A contains the $>'$ cycle $1 >' 2 >' 3 >' 1$. Then, from $e(1 > 2) = e(1 > 2 > 3) + e(1 > 3 > 2) + e(3 > 2 > 1)$ (and similar results for $e(2 > 3)$ and $e(3 > 1)$), we have,

$$1 >' 2 \Rightarrow b_3 b_{23} + b_2 b_{23} + b_2 b_{12} > b_3 b_{13} + b_1 b_{13} + b_1 b_{12}. \quad (6.1)$$

$$2 >' 3 \Rightarrow b_3 b_{23} + b_3 b_{13} + b_1 b_{13} > b_2 b_{23} + b_2 b_{12} + b_1 b_{12}. \quad (6.2)$$

$$3 >' 1 \Rightarrow b_1 b_{13} + b_2 b_{12} + b_1 b_{12} > b_3 b_{23} + b_2 b_{23} + b_3 b_{13}. \quad (6.3)$$

Adding inequalities (6.1) and (6.2), then (6.1) and (6.3), and finally (6.2) and (6.3) gives

$$\begin{aligned} b_3 b_{23} &> b_1 b_{12} > 0, \\ b_2 b_{12} &> b_3 b_{13} > 0, \\ b_1 b_{13} &> b_2 b_{23} > 0. \end{aligned}$$

Then vertical multiplication gives $b_1 b_2 b_3 b_{12} b_{13} b_{23} > b_1 b_2 b_3 b_{12} b_{13} b_{23}$, a contradiction. A similar contradiction results if we assume that A contains the $>'$ cycle $1 >' 3 >' 2 >' 1$. Therefore, A cannot contain any $>'$ cycle. \square

It is easily verified that if $|A| = 2$, then there cannot be a $>'$ cycle in B . If $|A| = 3$, it is not known whether B may contain a $>'$ cycle. The computations for this case are feasible, but exceedingly delicate in regard to the construction of a possible $>'$ cycle in B .

That no smaller example yields a $>'$ cycle within $A = \{1, 2, 3, 4\}$ was proved by an exhaustive computer search. For distinct i, j and k in A : let x_i be the number of elements of B covered only by i ; let x_{ij} be the number of elements of B covered by both i and j , and nothing else; let x_{ijk} be the number of elements of B covered just by i, j and k ; and let x_{1234} be the number of elements of B covered by all the elements of A . (In Figure 6.1, $x_1 = x_{13} = x_{24} = x_{123} = 1, x_{14} = 2, x_{234} = 5$, the remaining x 's equal 0, and $\sum x = 11$.) Complete enumeration, up to permutations on A , showed that no $>'$ cycle occurs when $\sum x \leq 10$. Moreover, the 15-element order of Figure 6.1 is the unique smallest case that yields a $>'$ cycle within A .

Chapter 7

Scheduling

One application of linear extensions is scheduling theory. Suppose that a set of “jobs” are to be processed by a set of “machines”, subject to some “precedence constraints”. The object is to produce a “schedule” which optimizes a given “measure of performance”. The processing order of the jobs on any particular machine is a linear extension of the precedence constraints. If the same linear extension of the precedence constraints is used on all machines, then the schedule is a “permutation schedule”.

We show that there is an optimal permutation schedule for the “ m -machine flow shop” with precedence constraints, provided that incomparable jobs have identical “processing times”. In this case, the problem is one of finding an “optimal” linear extension of the order. If the measure of performance is symmetric, then all permutation schedules are optimal.

7.1 Introduction

A general *job shop* problem consists of a set of n jobs J , which must be processed by a set of m machines M_1, \dots, M_m . Each job must be processed

exactly once by each machine, and in turn, each machine must process each job exactly once. Furthermore, a machine may process only one job at a time, and a job may be processed by only one machine at a time. For each pair of a job x and a machine M_i , the *processing time* $t(x, i)$, is the length of time required by machine M_i to process job x . All processing times are known, and are assumed to include any set-up time involved to prepare a machine for a job. Each job has a *technological constraint*, a linear ordering of the machines, which specifies the order in which the machines must process that job. In general, different jobs have different technological constraints. A special case of the general job shop problem is the *flow shop* problem, in which the technological constraints are the same for all jobs. In a flow shop the machines are numbered such that each job must be processed by the machines in the numerical order M_1, M_2, \dots, M_m . A scheduling problem may also include *precedence constraints*, a (partial) ordering of the jobs which any schedule must follow. That is, if x and y are two jobs with $x < y$, then on each machine, job x must be processed before job y . An *unconstrained* problem is one in which the order on the jobs is an antichain. Unless otherwise specified, problems are unconstrained: there are no restrictions on the order in which the jobs are processed.

Example 7.1 A woodworking shop produces three different toys: a puzzle; a car; and a duck. These are the jobs. The production involves four different processes: cutting the pieces; sanding; assembly; and painting. These are the machines. There may not be any actual "machines" involved; in this example the machines are person-tool combinations, i.e., a person with a saw, a hammer and glue, or a paintbrush. The order of processing is

different for the different jobs: the car is cut, sanded, assembled, then painted; the puzzle must be painted before it is assembled; the duck needs to be sanded after assembly, but before painting. We list the order of processing and processing times in a table.

job	processing order/time			
puzzle	cutting, 8	sanding, 7	painting, 5	assembly, 2
car	cutting, 4	sanding, 4	assembly, 3	painting, 3
duck	cutting, 6	assembly, 2	sanding, 10	painting, 6

This is a four-machine unconstrained job shop problem with three jobs.

Example 7.2 A four-machine unconstrained flow shop problem with two jobs.

job	processing order/time			
1	$M_{1,1}$	$M_{2,4}$	$M_{3,4}$	$M_{4,1}$
2	$M_{1,4}$	$M_{2,1}$	$M_{3,1}$	$M_{4,4}$

A *schedule* is a specification of how the jobs are to be processed by the machines, subject to the above restrictions. That is, a schedule specifies not only the order of processing of the jobs on each machine, but also the starting times of the jobs. (For a “regular” measure of performance it is sufficient to specify the processing order, as it may be assumed that all jobs are started as soon as possible.) A *permutation schedule* is a schedule in which the processing order of the jobs is the same on all of the machines. A schedule may be presented simply as a table listing the starting times of each job on each machine. We prefer to use a pictorial representation called a *Gantt Diagram*. (See Example 7.3.) The diagram consists of a horizontal

time axis and a row for each machine. In the row for any given machine, the time intervals when the jobs are being processed on that machine are blocked off and labelled. The advantage of the Gantt Diagram is that it is much easier to observe *idle time*, times when a machine is not processing any jobs; and the *waiting time* for each job, time when the job is not being processed by any machines.

Example 7.3 Three possible schedules for the flow shop problem of Example 7.2. The first two schedules are permutation schedules, the last is not.

time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
M_1	1	2													
M_2	1				2										
M_3						1			2						
M_4										1	2				

time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
M_1	2				1										
M_2						2	1								
M_3							2	1							
M_4								2			1				

time	1	2	3	4	5	6	7	8	9	10	11	12	
M_1	1	2											
M_2	1				2								
M_3							2	1					
M_4										2	1		

The objective in all scheduling problems is to find an *optimal* schedule, one which minimizes a given measure of performance. The *completion time*

C_x of a job x is the (actual) time when the last machine M_m , finishes processing job x , assuming that the processing starts at time zero. A *measure of performance* is a real-valued function of the completion times of the jobs. A *regular* measure of performance is one which is nondecreasing in all completion times. Most measures of performance, including all measures which we consider, are regular. Indeed, it is a strange measure of performance which "penalizes" a schedule for finishing some jobs earlier (than another schedule), without finishing any jobs later. Call a measure of performance M , *symmetric* if

$$M(C_{x_1}, C_{x_2}, \dots, C_{x_n}) = M(C_{x_{\pi(1)}}, C_{x_{\pi(2)}}, \dots, C_{x_{\pi(n)}})$$

for any permutation π of $\{1, 2, \dots, n\}$. Most frequently used measures of performance are of the form $F_{\max} = \max\{F_x(C_x) | x \in J\}$ or $\sum F = \sum_{x \in J} F_x(C_x)$, for some real valued functions F_x . If the functions F_x are all nondecreasing, then the measure of performance is regular. Two common measures of performance are $\bar{C} = 1/n \sum_{x \in J} C_x$, the *average completion time* and $C_{\max} = \max\{C_x | x \in J\}$, the *maximum completion time*. Average and maximum completion times are both regular, symmetric measures of performance. The jobs may have *due dates*: for each job, a time D_x is specified, by which the last machine should have finished processing that job. A *tardy* job is one whose completion time exceeds its due date. The *tardiness* of the job is $T_x = C_x - D_x$. If a job is completed before its due date, the tardiness is 0.

[Johnson 54] shows that there are optimal permutation schedules for the two-machine flow shop with a regular measure of performance; and

for the three-machine flow shop, for maximum completion time of a job. However, with more machines, or a measure of performance which is not regular, there need not be any optimal permutation schedules. Indeed, the schedules of Example 7.3 show that even a four-machine flow shop with only two jobs need not have an optimal permutation schedule for maximum (or average) completion time.

Even when an optimal permutation schedule exists, it may be difficult to find. Johnson found an optimal (permutation) schedule for the two-machine flow shop with respect to C_{\max} . However, the same problem is NP-complete on three (or more) machines, [Lenstra/Rinnooy Kan/Brücker 77]; or with precedence constraints, [Monma 80]. [Monma/Sidney 79] found an optimal solution for the two-machine flow shop with series-parallel precedence constraints with respect to the same measure C_{\max} . The two-machine flow shop problem with measure $\sum C_x$ was shown to be NP-complete in [Garey/Johnson/Sethi 76]. Little more is known about exact solutions.

7.2 Main Results

Theorem 7.1 (Optimal Permutation) *There is an optimal permutation schedule for the m -machine flow shop with precedence constraints, provided that*

$$t(x, i) = t(y, i),$$

for $i = 1, 2, \dots, m$; whenever x and y are incomparable.

The last condition in the theorem is referred to as the *identical processing time*, or IPT, assumption. Call jobs x and y equivalent if there is

a sequence $x = z_1, z_2, \dots, z_k = y$, such that z_j is incomparable to z_{j+1} , ($1 \leq j \leq k-1$). This defines an equivalence relation on the jobs, partitioning them into blocks B_1, B_2, \dots, B_b . The precedence order is a linear sum of these blocks. As a result, the processing orders of the jobs on different machines can differ only in the order of processing jobs within a block. Let $x, x' \in B_r, y \in B_s$, with $r \neq s$. Jobs contained in different blocks must be comparable. Thus, both x and x' are comparable to y . Assume, without loss of generality, that $x < y$. Now, suppose that $y < x'$. Since $x, x' \in B_r$, there exist $z_i \in B_r$ such that $x = z_1, z_2, \dots, z_k = x'$ and z_j and z_{j+1} are incomparable. Since all z_j are comparable to y , there exists j such that $z_j < y$, but $z_{j+1} > y$. Then $z_j < z_{j+1}$, a contradiction. Therefore, $x' < y$.

Proof (Optimal Permutation) We show that, for any schedule S , there is a permutation schedule S' which gives the same value of the measure of performance as S . Therefore, an optimal permutation schedule exists. The processing order of the jobs under the schedule S' is the processing order of the jobs on the last machine M_m , under S .

Let S be a schedule. Label the jobs in the single block B_r such that the order of processing on the last machine is x_1, x_2, \dots, x_k . Let $x_{j,i}$ denote the j th job (from B_r) processed by the i th machine. Thus, $x_{j,m} = x_j$. A new schedule is formed by scheduling x_j on M_i during the time when S had M_i processing the job $x_{j,i}$. It is possible to process x_j during this time as x_j and $x_{j,i}$ have identical processing times. This new ordering of the jobs on M_i observes the precedence constraints, as the ordering of the jobs on the last machine is valid. We need only verify that machine M_{i-1} finishes processing the job x_j before machine M_i starts. Under S ,

machine M_{i-1} finishes processing the j jobs $x_{1,i}, x_{2,i}, \dots, x_{j,i}$, before M_i starts processing $x_{j,i}$. Therefore, M_{i-1} has time to process the j jobs x_1, x_2, \dots, x_j , before M_i is to start processing x_j . This new schedule is a permutation schedule within the block B_r . As the processing on the final machine was not changed, this new schedule is equivalent, in terms of the measure of performance, to S . The jobs in the other blocks are reordered in a similar manner, producing the permutation schedule S' . \square

In the proof of the theorem, it is possible that rearranging the jobs may make it possible to complete some jobs sooner. When the jobs are reordered on the machines $M_1 - M_{m-1}$, there may be unnecessary idle time. It may occur that machine M_i is to process job x next, and that both the job and the machine are available, but processing does not start because, under the schedule S , processing started at a later time. For example, consider a two-machine flow shop with two jobs x, y , without precedence constraints, and with unit processing times. Suppose that the schedule S has M_1 processing y first, but M_2 processing x first. Then machine M_2 cannot begin job x until M_1 has processed both jobs: M_2 cannot begin job x before time $t = 2$. However, when the jobs are reordered on M_1 , the machine M_1 need only process job x before M_2 can begin processing job x . Therefore, machine M_2 could begin processing job x at time $t = 1$. Removing such unnecessary idle time may result in some jobs finishing sooner. If the measure of performance is regular, this produces a schedule which is certainly as good as, or possibly even better than, S . However, for a measure which is not regular, this need not be the case, decreasing some completion times may produce a schedule which is not even as good as S .

Let S be a permutation schedule for an m -machine flow shop which satisfies the IPT assumption. Since the jobs within a block have identical processing times, the jobs within a block may be permuted (respecting the precedence constraints) without changing the set of completion times for the jobs. Certainly, the completion times of individual jobs may change, but the set of completion times on that block remains the same. However, as the blocks are linearly ordered, jobs from different blocks cannot be permuted without violating the precedence constraints. Therefore, any two schedules can differ only in the processing order of the jobs within individual blocks. As a result, any two permutation schedules result in the same set of completion times for the jobs: they differ only in their assignment of these times to the jobs. This immediately yields the following corollary to the theorem.

Corollary 7.1 *All permutation schedules are optimal for the m -machine flow shop with precedence constraints and a symmetric measure of performance, when the IPT assumption holds.*

The well known *linear (maximum) assignment* problem requires finding the permutation of n items which minimizes the sum (maximum) of the costs of the assignments, where the cost of assigning the k th item to the j th position in the permutation is given for each k and j . Call such an assignment problem *ordered* if the permutation has to be consistent with an ordering of the items. That is, $x \leq y$ implies that x is assigned to an earlier position than y .

Theorem 7.2 (Linear Assignment) *The m -machine flow shop problem with precedence constraints, measure of performance $\sum F$ (F_{\max}) and the IPT assumption, reduces to b ordered linear (maximum) assignment problems defined on the blocks B_1, \dots, B_b .*

Proof (Linear Assignment) Theorem 7.1 implies that there is an optimal permutation schedule. Since any two permutation schedules differ only in their assignment of the (fixed) set of completion times to the jobs, the problem reduces to a linear assignment problem.

Consider a block B_r with k jobs. Let the fixed completion times for the block be $C_1 < C_2 < \dots < C_k$. Let $c_{xj} = F_x(C_j)$ be the cost of completing job x at time C_j , for $1 \leq j \leq k$ and $x \in B_r$. Then, finding the best order of processing the jobs in B_r , for a regular measure of performance of the form $\sum F = \sum F_x(C_x)$, clearly reduces to solving the ordered linear assignment problem,

$$\text{minimize } \sum_{x \in B_r} \sum_{j=1}^k c_{xj} Y_{xj},$$

subject to:

$$\sum_{x \in B_r} Y_{xj} = 1 \quad j = 1, 2, \dots, k; \quad (7.1)$$

$$\sum_{j=1}^k Y_{xj} = 1 \quad x \in B_r; \quad (7.2)$$

$$Y_{xi} \leq \sum_{j=1}^i Y_{xj} \quad \text{for } z \leq x, i = 1, 2, \dots, k; \quad (7.3)$$

$$Y_{xj} = 0 \text{ or } 1 \quad x \in B_r, j = 1, 2, \dots, k. \quad (7.4)$$

Constraint (7.3) ensures that the ordering of B_r by the optimal assignment is compatible with the precedence constraints. For performance measures of the form $F_{\max} = \max F_x(C_x)$, the formulation of the assignment problem is the same, except that the objective function is changed to $\min \max\{c_{xj}Y_{xj} | x \in B_r, j = 1, 2, \dots, k\}$.

Solving one assignment problem for each block determines the optimal processing order for the jobs within each block. Since the blocks are totally ordered, this induces an optimal permutation schedule for the original flow shop problem. \square

Johnson's algorithm finds an optimal permutation schedule for the two-machine unconstrained flow shop with the measure of performance, maximum completion time, in $O(n \log n)$ time. Monma and Sidney are able to find an optimal permutation schedule for the two-machine flow shop with series-parallel precedence constraints in $O(n \log n)$ time too. For the m -machine flow shop with a symmetric measure of performance, we need only find a total ordering of the jobs compatible with the precedence constraints to find an optimal schedule. This can be done in $O(n^2)$ time.

Corollary 7.2 *The least-cost-last algorithm finds an optimal permutation schedule for the m -machine flow shop with precedence constraints, with processing times that satisfy the IPT assumption, and with a regular measure of performance of the form $\max\{F_x(C_x) | x \in J\}$.*

Proof (Corollary 7.2) Any two permutation schedules have the same set of completion times. Let $C_1 < C_2 < \dots < C_n$ be the completion times. The *least-cost-last* algorithm, which bears a resemblance to a well

known single-machine scheduling algorithm (cf. [Lawler 73]) constructs an optimal permutation schedule as follows. The schedule is constructed in the reverse order. At each step we consider the set V , of jobs that are not yet scheduled, and determine which one to process last. Let U be the set of those jobs from V which, with respect to the precedence constraints, may be scheduled last. If $k = |V|$ then the completion time of the job that is processed last is C_k . Choose a job x from U , such that $F_x(C_k) \leq F_y(C_k)$ for all jobs $y \in U$. Job x is processed only after all other jobs in V are processed. Remove job x from the set V and repeat this procedure.

The proof that such a schedule is optimal is straightforward. It follows much the same argument as used to verify the original least-cost-last algorithm for single-machine scheduling (cf. [French 82], where it is commonly called "Lawler's algorithm"). \square

Corollary 7.3 *The precedence constrained m -machine flow shop problem, even with the IPT assumption satisfied, is NP-complete for the following regular measures of performance:*

- the total (average) weighted completion time ($\sum w_x C_x$);
- the total (average) tardiness ($\sum T_x$);
- the total number of tardy jobs ($\sum U_x$).

Proof (Corollary 7.3) Any single-machine scheduling problem with precedence constraints and identical processing times is a special one-machine flow shop problem in which the IPT assumption holds. Thus, the NP-completeness of the precedence constrained m -machine flow shop prob-

lem with IPT assumption, follows from the NP-completeness of the corresponding single-machine problem with precedence constraints and identical processing times. This was shown for: $\sum w_x C_x$ in [Lenstra 77], [Lawler 78] and [Lenstra/Rinnooy Kan 78]; for $\sum T_x$ in [Lenstra 77] and [Lenstra/Rinnooy Kan 78]; for $\sum U_x$ in [Garey/Johnson 76], [Lenstra 77] and [Lenstra/Rinnooy Kan 78].

We note that Theorem 7.1 and Corollary 7.3 also imply that the partially ordered linear assignment problem is NP-complete. Contrasting this, Lawler's original argument could also be used to show that the least-cost-last algorithm solves the partially ordered maximum assignment problem as long as the assignment costs are monotone nondecreasing, i.e., $j < k$ implies $c_{xj} < c_{xk}$ for every x . \square

Corollary 7.4 *The m -machine flow shop problem with a regular measure of performance of the form $\sum F_x(C_x)$ and the IPT assumption satisfied can be solved in polynomial time, if jobs within the same block are incomparable for every block.*

Proof (Corollary 7.4) If the jobs within a block are incomparable, constraint (7.3), in the linear assignment formulation in the proof of Theorem 7.2, is empty. That is, the flow shop problem reduces to a series of linear assignment problems. These problems are known to be solvable in polynomial ($O(n^3)$) time. \square

7.3 General Flow Shop Problems

In this section, applications of previous results for flow shop problems which do not satisfy the IPT assumption are considered.

In view of the difficulty of finding an optimal schedule, we may be willing to accept a schedule which is "almost" optimal. However, without knowing an optimal schedule, it is difficult to decide if a given schedule is close to begin optimal or not. One possibility is to define a similar problem, which is more readily solved, such that an optimal schedule for the new problem is no worse than an optimal schedule for the old problem. Given a scheduling problem not satisfying the IPT assumption, we modify the processing times of the jobs so that the IPT assumption is satisfied.

Corollary 7.5 *Given an m -machine flow shop problem with precedence constraints, processing times $t(x, i)$, and a regular measure of performance, define a new problem using the processing times*

$$t'(x, i) = \min\{t(y, i) | y \in B_r\} \quad \text{for } x \in B_r.$$

The cost of an optimal schedule for the original problem is not less than the cost of the optimal permutation schedule for the new problem.

Proof Let S be any schedule for the original problem; that is, using the original processing times. A schedule for the new problem can be formed from S by starting the processing of job x on machine M_i at the same time processing starts under the schedule S . This is possible as no processing times have been increased. Since no completion times have been increased, and the measure of performance is regular, this schedule is at least as

good as S . Therefore, an optimal schedule for the flow shop with the new processing times gives a lower bound for an optimal schedule for the flow shop with the original processing times. However, the new processing times were defined so that the new problem satisfies the IPT assumption. \square

Corollary 7.2 provides an effective procedure to compute the lower bound of Corollary 7.5, as long as the measure of performance has the form $\max\{F_x(C_x) | x \in J\}$. For other regular measures of performance we present a somewhat weaker but also easily computable lower bound.

Corollary 7.6 *Given an m -machine flow shop problem with precedence constraints, processing times $t(x, i)$, and a regular measure of performance of the form $\sum F_x(C_x)$, define a new problem with processing times*

$$t'(x, i) = \min\{t(y, i) | y \in B_r\} \quad \text{for } x \in B_r,$$

and with precedence constraints derived from the original ones by deleting all comparabilities between jobs within the same block. Then, the cost of an optimal schedule for the original flow shop is not less than the cost of the optimal permutation schedule for the new problem.

Proof Consider the intermediate problem with the new processing times $t'(x, i)$, and the original precedence constraints. By Corollary 7.5, the optimal value for the measure of performance for this intermediate problem represents a lower bound for the unknown optimal value for the original problem. Dropping some comparability relations represents a relaxation of this intermediate problem. Thus, the optimal objective value for the

relaxed problem of Corollary 7.6 represents a lower bound on the optimal value for the original flow shop problem. \square

Corollary 7.4 provides an effective procedure to compute the lower bound of Corollary 7.6.

7.4 Concluding Remarks

We have shown that if the IPT assumption is satisfied, then the set of completion times is fixed for the blocks. This property means that the precedence constrained m -machine flow shop with the IPT assumption satisfied is very closely related to single-machine scheduling problems. It may be that the completion times $C_1 < C_2 < \dots < C_n$ are such that the last machine M_m will have to be idle between some consecutive jobs from the same block. However, when this is not the case; that is, the completion times are such that M_m is never idle between jobs from the same block, then it is easily seen that the m -machine flow shop problem is equivalent to a collection of b , one-machine scheduling problems. The i th problem is defined on the block B_i , with the same performance measure, and the precedence constraints induced by B_i . This means that the NP-completeness results of Corollary 7.4 also apply when this special *no-wait* condition is satisfied. Furthermore, polynomial time algorithms available for single-machine scheduling problems with specially structured precedence constraints are directly applicable to the flow shop in this case. For example, the polynomial time solvability of the one-machine total weighted completion time ($\sum w_x C_x$) problem with series-parallel precedence constraints ([Lawler 78] and [Monma/Sidney 79]) means that, if the IPT assumption and the

7.4. CONCLUDING REMARKS

129

no-wait condition are satisfied, the total weighted completion time problem with series-parallel precedence constraints becomes effectively solvable in the m -machine flow shop too.

Chapter 8

Genus and Diagram Invariance

We investigate the problem of avoiding edge crossings when drawing the diagrams of ordered sets. A simple “lifting” construction is developed which, starting from a drawing, without edge crossings, of the covering graph of an order, produces a diagram, in which no edges cross. The construction shows that the “genus” of the diagram is equal to the “genus” of the order. This gives an example, indeed, the first nontrivial example, of a diagram invariant.

8.1 Planarity and Genus

When drawing a graph, it is preferable to avoid, where possible, edge crossings. The edge crossings make the graph difficult to read, and obscure its structure. A planar graph is one which may be drawn in the plane, without any edges crossing. Not all graphs are planar. For example, the graphs $K_{3,3}$ and K_5 are nonplanar. These are the minimal nonplanar graphs, in

the sense that any nonplanar graph “contains” one of these graphs. However, any (nonplanar) graph can be drawn without edge crossings, on some appropriate “surface”.

Start with the sphere $S = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}$. It is well known that drawing a graph without edge crossings on the sphere is equivalent to drawing the graph without edge crossings in the plane. Suppose that the graph cannot be drawn without edge crossings on the sphere. Edge crossings can be eliminated by “building bridges” on the sphere for edges to pass over other edges, avoiding intersection. These bridges are commonly referred to as *handles*, and differ from the common notion of a bridge in that edges are not restricted to traversing the “top” of the bridge only. In fact, it is possible for two edges to “cross” on a handle without any edge crossings, in the usual sense, by drawing the edges on opposite sides of the handle. Handles may also be drawn between the sphere and other handles, or between pairs of handles.

Every orientable surface is topologically equivalent to a sphere with handles. The number of handles is the genus of the surface. The sphere has genus zero. A torus, equivalent to a sphere with a single handle, has genus one. The genus of a graph is the minimal genus of a surface on which the graph can be drawn without edge crossings. In fact, if a graph can be drawn on a surface without edge crossings, then it can be drawn without edge crossings on any topologically equivalent surface.

Edge crossings are also undesirable when drawing the diagrams of ordered sets. We consider the problem of avoiding edge crossings in diagrams. A *planar order* is one whose diagram can be drawn (in the plane) without

edge crossings. The *genus* of an order is the minimal genus of a surface on which the order can be drawn without edge crossings. (When drawing diagrams on surfaces, in three dimensions, we compare the relative heights of the elements by comparing z -coordinates.) Drawing diagrams is somewhat more complicated than drawing graphs. This is due to the special requirements of the diagram: relative vertical placement of vertices, and monotonicity of edges.

8.2 Genus Zero Orders

The diagram of an order is a graph, the covering graph of the order. Trivially, a planar order has a planar covering graph. However, there are nonplanar orders having planar covering graphs. The cube, shown in Figure 8.1, is a nonplanar order which has a planar covering graph. In contrast to graphs, it is possible that the diagram of a nonplanar order may be drawn on the surface of the sphere without edge crossings. Indeed, the cube may be drawn on the surface of the sphere, without edge crossings, as shown in Figure 8.2.

This fact gives rise to the question - can every order with a planar covering graph be drawn without edge crossings on the surface of the sphere? Unfortunately, the answer is no.

Proposition 8.1 *The "double cube" (of Figure 8.3) cannot be drawn on the surface of the sphere, without edge crossings.*

Proof We describe why the diagram of the double cube cannot be drawn, without edge crossings, on the surface of the sphere. In view of the mono-

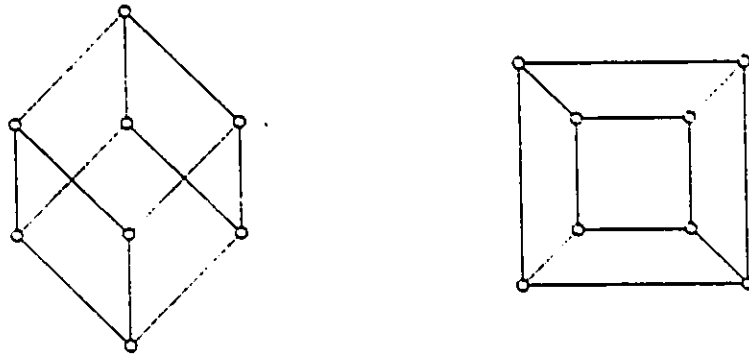


Figure 8.1: A nonplanar order with a planar covering graph

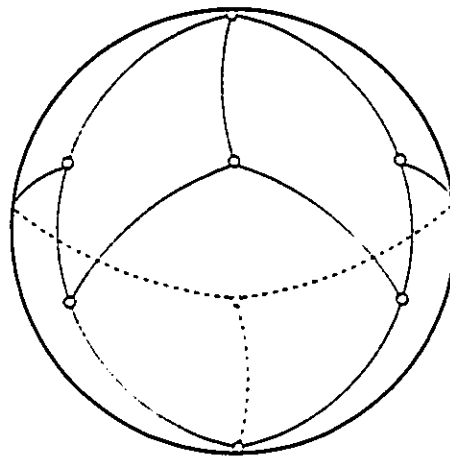
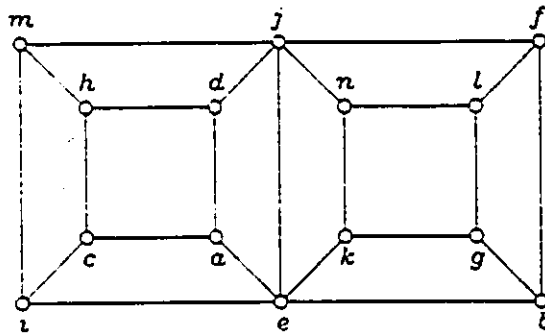
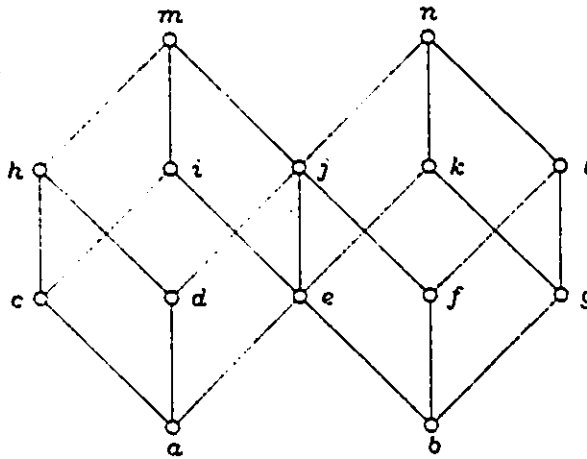


Figure 8.2: The cube on the sphere without edge crossings



The double cube has a planar covering graph but is not embeddable on the sphere

Figure 8.3: Double cube

tonicity of diagram edges, the only vertex that may be put at the north pole is one representing a maximal element. In particular, j cannot be put at the north pole. Dually, e , while having a smaller z -coordinate than j , cannot be put at the south pole. Let us suppose then that j and e are placed on the surface, both away from the poles, and joined by a monotonic arc rising from e to j .

Observe that j has two upper covers m, n , while e has two lower covers a, b . There are edge disjoint chains $m > i > e, n > j > e, j > d > a, j > f > b$. To avoid crossing edges, the closed curves $j < m > i > e$ and $j < n > j > e$ must be drawn so that one encloses the other in the sense that, either no monotonic path can be drawn from the south pole to n or j without crossing an edge, or no monotonic path can be drawn from the south pole to m or i without crossing an edge. If $j < n > j > e$ is enclosed by $j < m > i > e$, then no monotonic path can be drawn from b to either n or j without crossing an edge; if $j < m > i > e$ is enclosed by $j < n > j > e$ then no monotonic path can be drawn from a to either m or i . \square

However, the double cube can be drawn without edge crossings on a surface which is topologically equivalent to the sphere, as shown in Figure 8.4. This leads us to ask, if every order with a planar covering graph can be drawn without edge crossings on some surface of genus zero? The answer is yes.

Theorem 8.1 (Genus Zero) *Any ordered set which has a planar covering graph may be embedded, without edge crossings, on a surface of genus zero.*

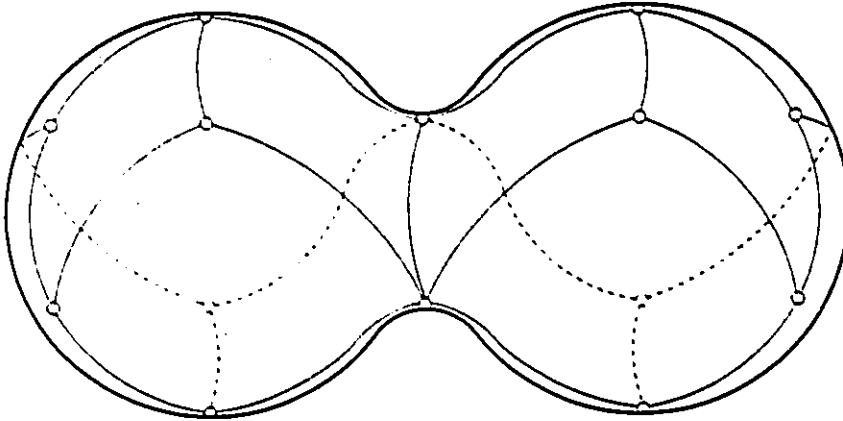


Figure 8.4: Embedding of double cube on a genus zero surface

Proof (Genus Zero) Suppose that the covering graph G of an ordered set P is planar. To produce a diagram, we start with a plane drawing of G and “lift” its vertices relative to their heights in P . Deforming the plane in the process of lifting produces a surface on which the diagram is drawn.

Suppose that G is drawn in the $z = 0$ plane of \mathbb{R}^3 , with straight edges which do not cross. Each vertex of G is lifted by one less than its height. That is, the vertex a with coordinates $(x, y, 0)$, is replaced by the vertex in \mathbb{R}^3 with coordinates $(x, y, h(a))$, where $h(a) = \text{height}(a) - 1$. For each pair of vertices joined by a covering edge, draw a straight edge in \mathbb{R}^3 joining the corresponding lifted vertices. This new edge is simply a lifted version of the old edge. This produces a drawing of the diagram of P on a surface, much like a topographic drawing of it, topologically equivalent to a hemisphere, which has genus zero. \square

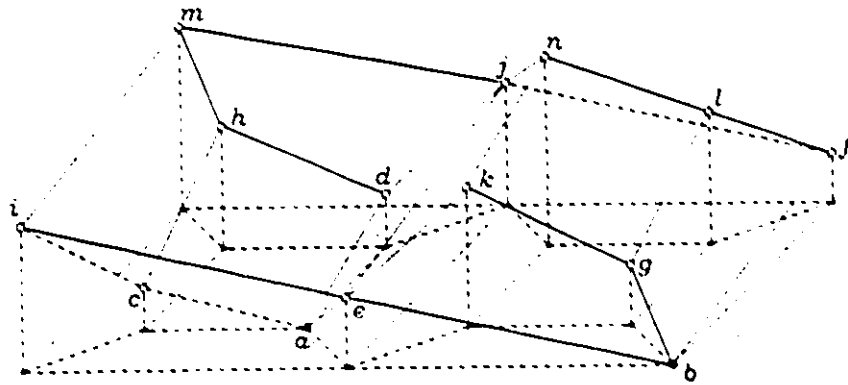


Figure 8.5: Lifting of the double cube

The lifting is illustrated for the double cube in Figure 8.5.

8.3 Higher Genus

Theorem 8.1 may be restated as: if the genus of the covering graph of an order is zero, then the genus of the order is zero. Using the polygon model of surfaces we are able to generalize this result to arbitrary genus.

Theorem 8.2 (Genus g) *The genus of any ordered set equals the genus of its covering graph.*

Proof (Genus g) The lifting construction of Theorem 8.1 may be generalized to a surface of any genus. Any surface has a polygon representation, according to which certain pairs of edges are identified. This polygon may be drawn in the $z = 0$ plane of \mathbb{R}^3 and, subsequently, lifted. This produces an embedding of the diagram of the order on a surface. This surface

an equivalent surface, which is realisable in \mathbb{R}^3 , it may not be possible to embed the order on such a surface.

We recall the definition of a surface (cf. [Henle 79]). First, a *triangle* T with vertices $v_i = (x_i, y_i, z_i)$, $i = 1, 2, 3$, in \mathbb{R}^3 consists of all points

$$v = (\lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3, \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3, \lambda_1 z_1 + \lambda_2 z_2 + \lambda_3 z_3),$$

where $\lambda_i \geq 0$, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. A (compact) *triangulated surface* is a topological space constructed from a finite number of triangles T_1, T_2, \dots, T_n , by identifying some of their common edges, such that any edge of a triangle is identified with at most one edge of another triangle. Thus, for a triangulated surface, we start with a set of disjoint triangles and then do the identifications. If two triangles have a common point, then this identification must be a result of a sequence of edge identifications of triangles between them. If triangles T_i and T_j share a common point of \mathbb{R}^3 which does not result from a pair of identified common edges of T_i and T_j , then this point is regarded as two distinct points, one belonging to T_i and the other to T_j . If this common point is a result of a pair of identified common edges, then it is considered to be a single point. A (compact) *surface* is a topological space homeomorphic to a (compact) triangulated surface. A surface may not have any realization at all in \mathbb{R}^3 . Any surface is homeomorphic to a polygon with identified edges. For example, a torus may be represented by a rectangle with parallel edges identified.

Given an ordered set P whose covering graph G has genus g , embed it (without edge crossings) on a surface with genus g . We consider this embedding on a polygon model (of this surface) with identified edges. Assume that this polygon is located in the $z = 0$ plane of \mathbb{R}^3 . In this polygon

model, vertices of G may be repeated, and its edges may be "cut" by a pair of identified edges on the boundary of the polygon. The boundary edges of the polygon as well as the (possibly) cut (covering) edges of G may all be taken as straight line segments, and all of these together form a plane graph. Actually, by adding "isolated" vertices to P , corresponding to the vertices of the polygon, and subdividing some covering edges of P by adding vertices, we may assume that the vertices of the polygon, together with the points at which an edge of G meets the polygon's boundary, are all included in the vertex set of P , hence G . (The addition of isolated vertices and subdivision vertices has no effect on the genus of P , or of G .) See, for example, Figure 8.6.

Subdivide the inner faces of this plane graph into triangles. Lift every vertex a with coordinates $(x, y, 0)$ of this plane graph to the vertex a' with coordinates $(x, y, h(a) - 1)$ and join pairs of these lifted vertices, with a straight line, provided the corresponding vertices in the $z = 0$ plane were endpoints of an edge. The inner part of the triangle $T = abc$ in the $z = 0$ plane corresponds to the lifted triangle $T' = a'b'c'$ in \mathbb{R}^3 accordingly, a triangular facet. We must show that this is a triangulated surface which, with appropriate identifications, is a surface of genus g on which a diagram of P is drawn.

Let a_1b_1 and a_2b_2 be a pair of boundary edges of the polygon in the $z = 0$ plane for which a_1 is to be identified with a_2 and b_1 with b_2 . Now, $a'_1b'_1$ and $a'_2b'_2$ are "broken lines" in \mathbb{R}^3 . According to the construction, corresponding pairs of vertices on these two "broken lines" have the same z -coordinate. To identify $a'_1b'_1$ with $a'_2b'_2$ we consider every triangular facet

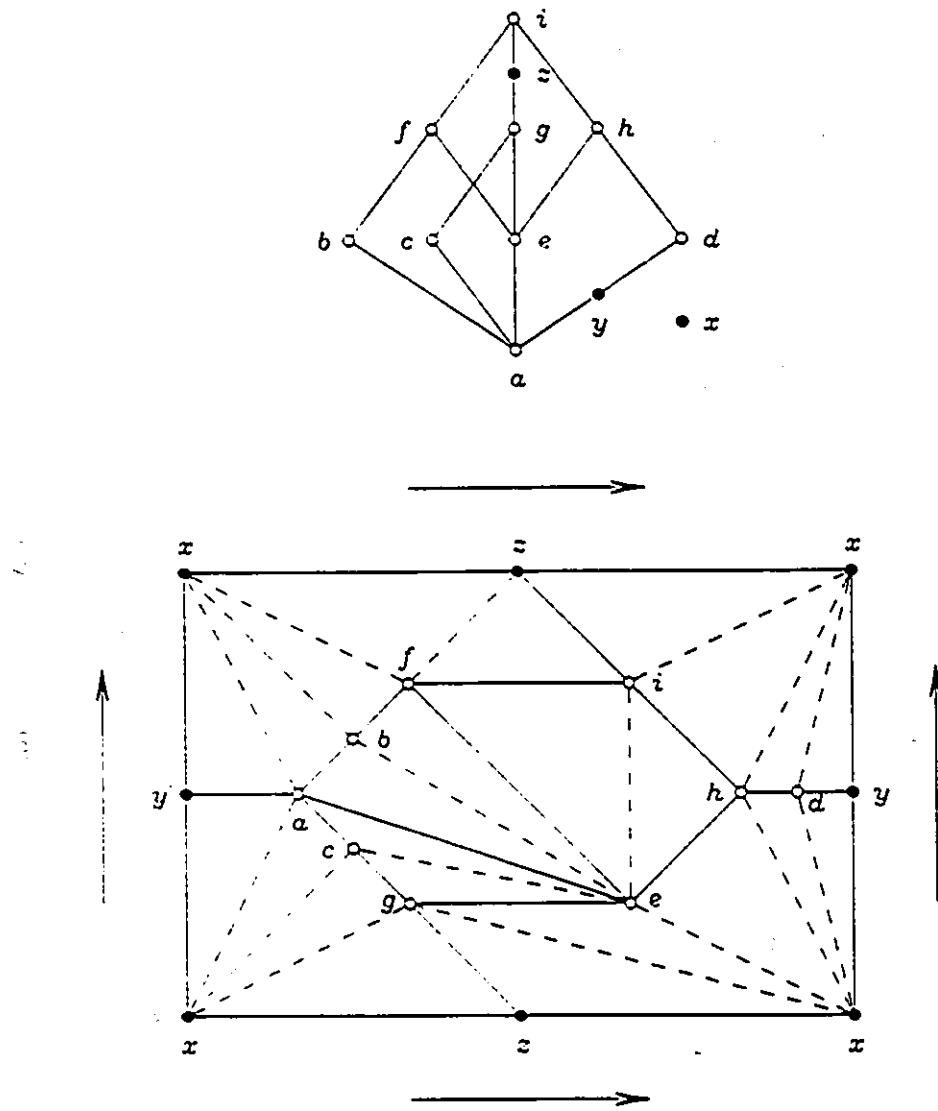


Figure 8.6: Polygon model of surface for genus one order

T' in the lifted graph with one side $a'_2b'_2$ or one vertex a'_2 or b'_2 . Identify each point on the edge $a'_2b'_2$ with its corresponding point on the edge $a'_1b'_1$ having the same z -coordinate. In this way, each point on $a'_2b'_2$ is moved horizontally to its corresponding point on $a'_1b'_1$ with the same z -coordinate. Each other edge of T' ending in a'_1 or b'_1 is associated with the edge in \mathbb{R}^3 joining the unmoved vertex and a'_2 or b'_2 , respectively. Any edge crossings that appear with the identifications can be avoided with a little "movement". Since there are only a finite number of edges and vertices, any such undesired edge crossings may be removed by moving a vertex slightly. We repeat this process for every pair of boundary edges of the polygon in the $z = 0$ plane which are to be identified. In this way we produce a surface with genus g and an embedding of P on it. This completes the proof. \square

The previous theorem shows that the genus of an ordered set depends only on the genus of its covering graph. An immediate consequence of this is the following theorem.

Theorem 8.3 (Diagram Invariant) *The genus of an ordered set is a diagram invariant.*

Another interesting consequence is,

Theorem 8.4 (Decision) *For a fixed integer g , the problem of deciding if the genus of ordered set is at most g , can be settled in polynomial time.*

Proof (Decision) The key point is the result in [Filotti/Miller/Reif 79]. They describe an $O(n^{O(g)})$ algorithm which given a graph G and a

positive integer g , finds an embedding of G , without edge crossings, on a surface of genus g , if such an embedding exists. \square

Chapter 9

Cataloguing Ordered Sets

There are numerous reasons why having a list of all orders with n or fewer elements would be very desirable. We may seek an order satisfying some specific property, possibly to prove or disprove some conjecture. When seeking such an order, it might be easier to search a list of orders, rather than trying to construct an order of the desired form. A list may also be used to prove that no such orders exist, at least not with less than $n + 1$ elements. If orders with the given property exist, a list would allow us to determine the prevalence of such orders among all orders. It is also possible that a study of such a list would suggest new properties, ones not before apparent.

Our interest in this subject resulted from a question about the prevalence of orders having no irreducible elements. The interest in this question stems from a result in [Rutkowski 89], which shows that there are exactly eleven orders, with ten or fewer elements, which have no irreducible elements, but have the fixed point property. An element is called *irreducible* if it has a single upper cover, or a single lower cover. An order is

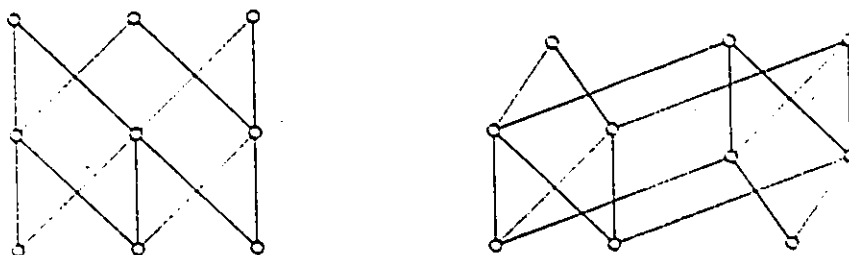


Figure 9.1: Orders without irreducible elements

dismantlable if it can be reduced to a singleton by successively removing irreducible elements. An order is said to have the *fixed point property* if every isotone mapping of the order to itself maps some element to itself. In [Rival 76], it was proved that dismantlability implies the fixed point property, but that the converse is not true. It is therefore interesting to know how common it is for nondismantlable orders to have the fixed point property. Since the absence of irreducible elements in a order implies that the order is nondismantlable, it is also natural to ask how many orders without irreducible elements have the fixed point property. Figure 9.1 shows two orders without irreducible elements. Both of these orders have the fixed point property.

While we originally considered finding only orders without irreducibles, our methods extend easily to the problem of finding all orders of a given size. This problem has been considered by many people. Most recently, [Chaunier/Lygerōs 92] computes that the number of thirteen-element orders is 33 823 827 452.

We consider the general problem first.

9.1 Generating Lists of Orders

It is important to understand the difference between labelled and unlabelled orders. In a labelled order, the elements are identified individually by means of labels. Whereas, in an unlabelled order, the individual elements are treated anonymously: only the general structure of the order is considered. In an unlabelled order it is not possible to ask questions about specific elements, such as

- is $a < b$?
- how many upper covers does a have?
- is a minimal?

However, any other order theoretic property not concerned with a specific element can be considered. This includes such things as: number of linear extensions; height and width; dimension; planarity and genus; presence of irreducible elements. The great advantage of unlabelled orders is that their number is much smaller than the number of labelled orders. The reason for this is simple, for an n -element unlabelled order, there are $n!$ different labellings. (Figure 9.2 shows three of the $24 = 4!$ possible labellings of the unlabelled, 4-element order $2 * 2$.) These labellings can yield up to $n!$ "different" labelled orders. However, these $n!$ different orders are all essentially the same: none of the basic properties of the order vary under the different labellings. It is possible that different labellings do not produce different orders. If two labellings are isomorphic they yield the same order. For example, in Figure 9.2, labellings (a) and (b) are

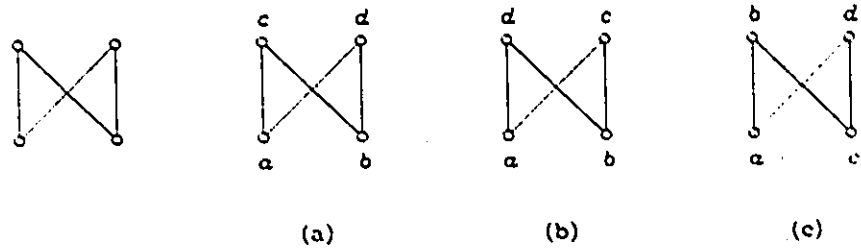


Figure 9.2: Several possible labellings of an unlabelled order

isomorphic, all comparabilities are the same, the labelled orders are, in fact, the same order. However, (c) is a different order since it includes the relation $a < b$ which does not occur in (a) and (b). In general, the number of automorphisms of an order is small. In fact, many orders have only the trivial automorphism. Thus, the number of different labelled orders is usually quite large.

Our objective is to generate the list of all n -element orders. For reasons made clear by the previous paragraph, we would like to consider unlabelled orders only. Unfortunately, our methods of storing and presenting orders inherently label the elements of the order. The solution is to consider labelled orders, but to define two labelled orders to be equivalent if their corresponding unlabelled orders are the same. This defines an equivalence relation on the labelled orders. To choose a representative from each equivalence class we use the idea of canonical orders, presented in [Colbourn/Read 79].

The orders are formed on the set $\{1, \dots, n\}$. To limit the number of orders generated, only those orders for which the natural ordering is a linear extension are considered. The orders are stored as an $n \times n$ upper triangular

matrix of 0s and 1s, representing the covering relations of the order. The entry at (a, b) is 1 if and only if a is covered by b . The comparabilities of the order are recorded similarly. These matrices are referred to as the covering and comparability matrices, respectively.

The elements of these (upper triangular) matrices are ordered lexicographically; that is, in the order

$$(1, 2), \dots, (1, n), \dots, (k, k+1), \dots, (k, n), \dots, (n-2, n), (n-1, n),$$

forming a $n(n-1)/2$ -element vector of 0s and 1s. This vector may be viewed as an $n(n-1)/2$ -digit binary number. The orders are generated in order of increasing value of this binary number: in the order,

$$\begin{aligned} & \text{antichain,} \\ & n-1 \prec n, \\ & n-2 \prec n, \\ & n-2 \prec n, n-1 \prec n, \\ & n-2 \prec n-1, \\ & n-2 \prec n-1, n-1 \prec n, \\ & n-2 \prec n-1, n-2 \prec n, \\ & n-3 \prec n, \\ & \vdots \\ & 1 \prec 2, 1 \prec 3, 1 \prec 4, \dots, 1 \prec n. \end{aligned}$$

Initially the vector is 0, the antichain. The last (rightmost) 0 in the vector is found, and tested to see if it can be set to 1. If it is possible to do so, then this element is set to 1 and the remainder of the vector (to the right) is cleared; otherwise, the next 0 is found and tested.

If the 0 is at position (a, b) , it is necessary to decide if the covering relation $a \prec b$ can be added. There are two possible problems to consider. Firstly, if $a < b$ already occurs in the order, then $a \prec b$ is not a valid covering edge. Secondly, adding $a \prec b$ might destroy some previously ex-

isting covering edge. In fact, the first problem cannot occur. Suppose that $a \prec b$ is to be added, but the relations $a < c < b$ are already present. We may assume, without loss of generality, that $c \prec b$. As natural numbers, $a < c < b$. Therefore, (a, b) is to the left of (c, b) and so, when the relation $a \prec b$ is added, the relation $c \prec b$ is removed. Thus, $a \prec b$ is a valid covering relation. The second problem is possible. That is, adding a new covering relation may destroy a previously existing one. Suppose that there exists c , with $c \prec b$, and $c < a$. Adding the covering relation $a \prec b$ gives $c < a < b$, so $c \prec b$ is no longer a covering relation. Thus, the covering relation $a \prec b$ may be added if and only if there is no c with $c < a$ and $c \prec b$.

Each time a new covering edge is added, a new order is produced. This order must either be added to the list, or rejected because it is equivalent to some order already in the list. Comparing an order with all orders in the list, to see if it is a relabelling of an order previously found, is very time consuming. This problem is solved by Colburn and Read through the use of "canonical orders". Linearly order the above-diagonal elements of the covering matrix. For reasons explained later, we order the elements lexicographically, considering the *second* component first; that is, in the order

$$(1, 2), (1, 3), (2, 3), (1, 4), (2, 4), (3, 4), \dots, (n-3, n), (n-2, n), (n-1, n).$$

This forms an $n(n-2)/2$ -digit binary number, the *value* of the order. Among all orders in an equivalence class there is one with minimum value. Call this order *canonical*, and let it be the representative of its equivalence class. To determine if an order should be added to the list, it is only necessary to test if the order is canonical. To test the order's canonicity, other

labellings of the order are tested, to see if there is a labelling with a lower value. If a labelling giving a lower value is found, the order is noncanonical and is discarded; otherwise, if all other labellings are tested and none gives a lower value, then the order is added to the list.

To test alternate labellings of the order, we consider permutations of the elements of the order. Initially, only permutations which swap a pair of elements are considered. (The elements swapped are necessarily incomparable.) The number of such permutations is small, relative to the total number of permutations, and these permutations are easily manipulated. Yet, these permutations are very efficient in eliminating noncanonical orders. In practice, these permutations eliminate between 70% and 90% of the noncanonical orders.

If the canonicity of the order is still undetermined, the remaining permutations of the order must be tested. Permutations are formed by selecting an element to replace 1, then selecting an element to replace 2, etc. When forming permutations we need only continue as long as the value of the new labelling is less than or equal to the value of the original labelling. Our ordering of the elements of the covering matrix yields the most significant digits of the binary number first, so we may decide if the new value exceeds the current one. As soon as this becomes apparent, we stop: no permutation completing the current initial sequence will prove that the order is not canonical. Our ordering of the covering matrix is very efficient: when the k th element of the permutation is chosen, the next $k - 1$ elements in the binary vector are determined. This fills in the binary vector very quickly, allowing us to abandon useless permutations sooner.

There is no point in testing permutations which correspond to automorphisms of the order: they yield the same order, with the same value. Most such permutations are avoided by the use of "identical" elements. Call a pair of elements of the order *identical* if they have the same upper and lower covers. A set of elements are identical if and only if they form an antichain which is an autonomous set. Any permutation which permutes only identical elements is an automorphism. To avoid such automorphisms, we order the elements of each identical set, and require all permutations to conform to this ordering.

This procedure is still very time consuming and can only be used for small values of n . Here are our humble results. (computations performed on a 80486 PC compatible running at 66 MHz.)

n	orders	time required	
4	16		
5	63	0.1	sec.
6	318	0.7	sec.
7	2045	14.9	sec.
8	16999	8.0	min.
9	183231	5.1	hrs.

9.2 Orders Without Irreducibles

We now describe the algorithm used to find all orders without irreducible elements. The algorithm is a modified version of the general algorithm. It would be very inefficient to produce all orders then decide which ones are irreducible. Therefore, the algorithm is modified so that only orders without irreducible elements are constructed. This is done by adding covering relations in such a way that no element is allowed to have a single upper, or a single lower cover. There are two instances where we must consider if

a covering edge can, or must be added. The first is when we must decide if the rightmost 0 (at position (a, b) in the vector) can be changed to a 1. In addition to deciding if the covering edge $a < b$ renders any other covering edges nonessential, we must avoid introducing irreducible elements. If $b = a + 1$ has no lower covers, then b cannot cover a . Among $1 \dots a - 1$, b has no lower covers, since there are no other possible lower covers for b , and b cannot have only one lower cover, it must not have any lower covers, to avoid an irreducible element.

The covering edge $a < b$ is added, if possible. Otherwise, the next rightmost 0 is found and tested. The second instance occurs after the covering edge $a < b$ has been added. Previously, in the general algorithm, the rest of the vector was cleared. It may not be possible to do that now, without introducing irreducible elements. After (a, b) has been set, it is necessary to test each element (c, d) , which is to the right of (a, b) in the vector. The following tests are used, (in addition to the usual considerations of nonessential edges):

- if $d = c + 1$ has a single lower cover then d must cover c — among $1 \dots c - 1$, d has only one lower cover, the only possible second lower cover is c ;
- if $d = c + 1$ has no lower covers then d cannot cover c — there is no possible second lower cover for d ;
- if $d = n$ and c has no upper covers then d cannot cover c — there is no possible second upper cover for c ;
- if $d = n$ and c has only one upper cover then d must cover c — d is

the only possible second upper cover for c .

If (c, d) may be cleared then it is done. If (c, d) cannot be cleared, then it is set, if possible. Otherwise, the order being constructed cannot be completed to an order without irreducibles, and is abandoned. Continuing on, the rightmost 0, to the left of (c, d) is found and tested to see if it may be set.

The fact that the number of orders without irreducibles is much smaller than the total number of orders allows a more efficient method for avoiding multiple labellings of an order. As observed before, if the elements of the covering matrix are ordered lexicographically, then the orders are produced in increasing order of value. Thus, the canonical order from any equivalence class, having lowest value, is found before all other orders in its class. This fact can be used to avoid alternate labellings of the order, by employing a method similar to a well known prime number sieve, commonly called the Sieve of Eratosthenes. When a canonical order is produced, all other labellings, and their associated values, are found. The values are stored in an ordered list, by means of a binary tree. When a new order is produced, to test its canonicity, it is sufficient to search for its value in the list. The order is canonical if and only if its value is not contained in the list. The advantage of this method is that permutations are formed only for canonical orders, rather than all orders. As the equivalence classes are very large, this represents a significant improvement in performance. The drawback with this method is the large amount of memory required to store the list of values. The size of this list makes it unsuitable for use when producing a list of all orders on n elements.

We are only interested in connected orders, but producing only connected orders is very difficult. However, many disconnected orders are easily avoided by producing orders which have no isolated elements. This is accomplished by requiring each element to have at least two upper, or two lower covers. The following tests are applied when considering if the covering relation $a \prec b$ must be added:

- if a has no lower covers and $b = n - 1$, then b must cover a ;
- if $b = a + 2 \geq n - 1$ has no lower covers, then b must cover a — since b cannot have two upper covers, it must have two lower covers, the only remaining possibilities are a and $a + 1$.

It is easily seen that this will, in fact, ensure that the order does not have any components with less than four elements. However, disconnected orders may still be produced, so the connectivity of the orders must be tested.

Our results are as follows,

n	orders	time required
4	1	
5	2	
6	7	
7	25	0.1 sec.
8	123	1.1 sec.
9	792	20.0 sec.
10	6965	13.1 min.
11	84124	10.5 hrs.

An interesting addendum, contrasting our results, is a subsequent result in [Schröder 92], which proves that the number of n -element orders without irreducible elements, but which have the fixed point property, is very large.

Bibliography

- [1] M. Aigner (1985) A note on merging, *Order* 2, 257–264.
- [2] M. D. Atkinson and H. W. Chang (1985) Extensions of partial orders of bounded width, *Congressus Numerantium* 52, 21–35.
- [3] M. D. Atkinson and H. W. Chang (1987) Computing the number of mergings with constraints, *Information Processing Letters* 24, 289–292.
- [4] M. D. Atkinson (1989) On computing the number of linear extensions of a tree, *Order* 7, 2^o–25.
- [5] I. Beck, Extension of partial orders, preprint.
- [6] I. Beck, On linear extensions of orders, preprint.
- [7] R. Bonnet and M. Pouzet (1982) Linear extensions of ordered sets, in: *Ordered Sets* (I. Rival, ed.), Reidel, 125–170.
- [8] G. Brightwell (1989) Semiorders and the 1/3–2/3 conjecture, *Order* 5, 369–380.
- [9] G. Brightwell, P. C. Fishburn and P. Winkler (1990) Interval orders and linear extension cycles, preprint.

- [10] G. Brightwell and P. Winkler (1991) Counting Linear Extensions, *Order* 8, 225–242.
- [11] C. Chaunier and N. Lygerōs (1992) Progrès dans l'énumération des posets, *C. R. Acad. Sci. Paris*, t. 314, Série I, 691–694.
- [12] C. Chaunier and N. Lygerōs (1992) The number of orders with thirteen elements, *Order* 9, 203–204.
- [13] C. J. Colbourn and R. C. Read (1979) Orderly Algorithms for Generating Restricted Classes of Graphs, *J. Graph Theory* 3, 187–195.
- [14] J. C. Culberson and G. J. E. Rawlins (1991) New results from an algorithm for counting posets, *Order* 7, 361–374.
- [15] R. R. Davidson and R. E. Odeh (1972) Some inconsistencies in judging problems, *J. Comb. Theory Ser. A* 13, 162–169.
- [16] G. Di Battista, W. P. Liu and I. Rival (1990) Bipartite graphs, upward drawings and planarity, *Information Processing Letters* 36, 317–322.
- [17] G. Di Battista and R. Tamassia (1988) Algorithms for plane representations of acyclic diagrams, *Theoret. Comp. Sci.* 61, 175–198.
- [18] R. P. Dilworth (1950) A decomposition theorem for partially ordered sets, *Ann. of Math.* 51, 161–166.
- [19] M. Dyer, A. Frieze and R. Kannan (1989) A Random Polynomial Time Algorithm For Approximating The Volume of Convex Bodies, *Proc 21st ACM Symp. on the Theory of Computing*, 375–381.

- [20] P. Edelman, T. Hibi and R. P. Stanley (1989) A Recurrence for Linear Extensions, *Order* 6, 15–18.
- [21] M. H. El-Zahar (1989) Enumeration of ordered sets, in: *Algorithms and Order* (I. Rival, ed.), Kluwer, 327–352.
- [22] K. Ewacha, P. C. Fishburn and W. V. Gehrlein (1990) Linear extension majority cycles in height-1 orders, *Order* 6, 313–318.
- [23] K. Ewacha, W. Li and I. Rival (1991) Order, genus, and diagram invariance, *Order* 8, 107–113.
- [24] K. Ewacha, I. Rival and G. Steiner (1990) Permutation schedules for flow shops with precedence constraints, *Opns. Res.* 38, 1135–1139
- [25] K. Ewacha, I. Rival and N. Zaguia (1993) Approximating the number of linear extensions, preprint.
- [26] K. Ewacha, I. Rival and N. Zaguia (1993) Unimodality, linear extensions and width two orders, preprint.
- [27] I. S. Filotti, G. L. Miller and J. Reif (1979) On determining the genus of a graph in $O(V^{O(g)})$ steps, *Proc. Eleventh ACM Symp. Th. Computing*, Atlanta.
- [28] P. C. Fishburn (1974) On the family of linear extensions of a partial order, *J. Combin. Theory* 17, 240–243.
- [29] P. C. Fishburn (1974) On the sum-of-Ranks winner when losers are removed, *Disc. Math.* 8, 25–30.

- [30] P. C. Fishburn (1974) Paradoxes of voting, *Amer. Pol. Sci. Rev.* **68**, 537-546.
- [31] P. C. Fishburn (1976) On linear extension majority graphs of partial orders, *J. Combin. Theory* **21**, 65-70.
- [32] P. C. Fishburn (1984) A correlational inequality for linear extensions of a poset, *Order* **1**, 127-137.
- [33] P. C. Fishburn (1986) Proportional transitivity in linear extensions of ordered sets, *J. Combin. Theory Ser. B* **41**, 48-60.
- [34] P. C. Fishburn and W. T. Trotter (1991) Linear extensions of semiorders: a maximization problem, *Disc Math* **97**, (proof).
- [35] S. Foldes, I. Rival and J. Urrutia (1992) Light sources, obstructions, and spherical orders, *Disc. Math.* **102**, 13-23.
- [36] D. J. Foulis (1970) *Empirical Logic*, University of Massachusetts.
- [37] M. Fredman (1976) How good is the information theory bound in sorting?, *Theoret. Comput. Sci.* **1**, 355-361.
- [38] S. French (1982) *Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop*, E. Horwood Publ., John Wiley, N.Y.
- [39] B. Ganter, G. Häfner and W. Poguntke (1987) On linear extensions of orders with a symmetry, *Disc. Math.* **63**, 153-156.
- [40] M. R. Garey and D. S. Johnson (1976) Scheduling tasks with nonuniform deadlines on two processors, *J. Assoc. Comput. Mach.* **23**, 461-467.

- [41] W. V. Gehrlein and P. C. Fishburn (1989) Linear extension majority cycles for small ($n \leq 9$) partial orders, *Comput. Math. Appl.* **20**, 41–44.
- [42] W. V. Gehrlein, An algorithm for generating the complete set of nonisomorphic partial orders, preprint.
- [43] J. Ginsburg, I. Rival and B. Sands (1986) Antichains and finite sets that meet all maximal chains, *Can. J. Math.* **XXXVIII**, 619–632.
- [44] G. H. Gonnet and J. I. Munro (1986) Heaps on heaps, *SIAM J. Comput.* **15**, 964–971.
- [45] R. L. Graham (1982) Linear extensions of partial orders and the FKG inequality, in: *Ordered Sets* (I. Rival, ed.), Reidel, 213–236.
- [46] M. Henle (1979) *A Combinatorial Introduction to Topology*, Freeman.
- [47] R. Jégou, R. Nowakowski and I. Rival (1987) The diagram invariant problem for planar lattices, *Acta Sci. Math. Szeged* **51**, 103–121.
- [48] S. M. Johnson (1954) Optimal two- and three-stage production schedules with setup times included, *Naval Res. Logist. Quart.* **1**, 61–68.
- [49] J. Kahn and M. Saks (1984) Balancing poset extensions, *Order* **1**, 113–126.
- [50] J. Kahn and N. Linial (1988) Balancing extensions via Brunn-Minkowski, preprint.
- [51] A. Karzanov and L. Khachiyan (1991) On the conductance of order Markov chains, *Order* **8**, 7–15.

- [52] D. Kelly (1985) Comparability graphs, in: *Graphs and Orders* (I. Rival, ed.), Reidel, 3–40.
- [53] D. Kelly (1986) Invariants of finite comparability graphs, *Order* 3, 155–158.
- [54] H. A. Kierstead and W. T. Trotter (1989) The number of depth-first searches of an ordered set, *Order* 6, 295–303.
- [55] S. S. Kislitsyn (1962) More precise definition of estimate of least mean number of comparisons necessary for complete ordering of finite totality (translation).
- [56] S. S. Kislitsyn (1966) Maximal k -decomposable sets of permutations of degree n , *Sib. Matem. Zh.* 7, 826–831 (translation).
- [57] S. S. Kislitsyn (1967) Partially ordered sets and k -decomposability, *Sib. Matem. Zh.* 8, 1197–1201 (translation).
- [58] S. S. Kislitsyn (1968) A finite partially ordered set and its corresponding set of permutations, *Mat. Zametki* 4, 511–518 (translation).
- [59] S. S. Kislitsyn (1968) A problem in combinatorial analysis, *Mat. Zametki* 3, 15–20 (translation).
- [60] D. Klarner (1986) The number of tiered posets modulo six, *Disc. Math.* 62, 295–297.
- [61] D. A. Klarner (1969) The number of graded partially ordered sets, *J. Comb. Theory* 6, 12–19.

- [62] D. Kleitman and B. Rothschild (1970) The number of finite topologies, *Proc. Amer. Math. Soc.* 25, 276–282.
- [63] D. Kleitman and B. Rothschild (1975) Asymptotic enumeration of partial orders on a finite set, *Trans. Amer. Math. Soc.* 205, 205–220.
- [64] G. Kreweras (1979) Sur les extensions linéaires d'une famille particulière d'ordres partiels, *Disc. Math.* 27, 279–295.
- [65] G. Kreweras (1985) Denombrement des ordres étages, *Disc. Math.* 53, 147–149.
- [66] E. L. Lawler (1973) Optimal sequencing of a single machine subject to precedence constraints, *Mgmt. Sci.* 19, 544–546.
- [67] J. K. Lawler (1978) Sequencing jobs to minimize total weighted completion time subject to precedence constraints, *Ann. Disc. Math.* 2, 75–90.
- [68] J. K. Lenstra, A. H. G. Rinnooy Kan and P. Brücker (1977) Complexity of machine scheduling problems, *Ann. Disc. Math.* 1, 343–362.
- [69] J. K. Lenstra and A. H. G. Rinnooy Kan (1978) Complexity of scheduling under precedence constraints, *Opns. Res.* 26, 22–35.
- [70] J. K. Lenstra (1977) Sequencing by enumerative methods, *Math. Centr. Tracts* 69, Mathematisch Centrum, Amsterdam.
- [71] N. Linial (1984) The information-theoretic bound is good for merging, *SIAM J. Comput.* 13, 795–801.

- [72] L. Lovász and M. Simonovits (1990) The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume, *Proc. 31st IEEE Symposium on Foundations of Computer Science*, 346–355.
- [73] H. M. MacNeille (1937) Partially ordered sets, *Trans. A. M. S.* 42, 416–460.
- [74] J. M. Miller and G. C. Stockman (1989) Determining search space size in precedence graphs, Submitted to 1st Great Lakes Conference.
- [75] J. M. Miller and G. C. Stockman (1990) On the number of linear extensions in a precedence graph, *IEEE International Robotics and Automation Conference*.
- [76] R. H. Möhring (1984) Almost all comparability graphs are UPO, *Disc. Math.* 50, 63–70.
- [77] R. M. Möhring (1985) Algorithmic aspects of comparability graphs and interval graphs, in: *Graphs and Orders* (I. Rival, ed.), Reidel, 41–101.
- [78] C. L. Monma (1980) Sequencing to minimize the maximum job cost, *Opns. Res.* 28, 942–951.
- [79] C. L. Monma and J. B. Sidney (1979) Sequencing with series-parallel precedence constraints, *Math. Opns. Res.* 4, 215–224.
- [80] M. Pouzet and I. Rival (1988) Is there a diagram invariant? *Disc. Math.* 73, 181–188.

- [81] R. W. Quackenbush (1982) Enumeration in classes of ordered structures, in: *Ordered Sets* (I. Rival, ed.), Reidel, 523-554.
- [82] R. C. Read (1978) Every one a winner, *Ann. Disc. Math.* 2, 107-120.
- [83] K. Reuter and I. Rival (1989) Genus of orders and visibility, *16th International Workshop Graph-Theor. Concepts Comp. Sci. 1990*, to appear.
- [84] I. Rival (1976) A fixed point theorem for finite partially ordered sets, *J. Combin. Theory, A* 21, 309-318.
- [85] I. Rival (1983) Optimal linear extension by interchanging chains, *Proc. A. M. S.* 89, 387-394.
- [86] I. Rival (1984) Linear extensions of finite ordered sets, *Ann. Disc. Math.* 23, 355-370.
- [87] I. Rival (1985) The diagram, in: *Graphs and Order* (I. Rival, ed), Reidel, 103-133.
- [88] I. Rival (1989) Graphical data structures for ordered sets, in: *Algorithms and Order* (I. Rival, ed.), Kluwer, 3-31.
- [89] I. Rival (1993) Reading, drawing, and order, in: *Algebras and Orders* (I. G. Rosenberg and G. Sabidussi, eds.), Kluwer, 359-404.
- [90] I. Rival and J. Urrutia (1988) Representing orders on the plane by translating convex figures, *Order* 4, 319-339.
- [91] I. Rival and N. Zaguia (1985) Antichain cutsets, *Order* 1, 235-347.

- [92] I. Rival and N. Zaguia (1987) Greedy linear extensions with constraints, *Disc. Math.* **63**, 249-260.
- [93] I. Rival and S. Zhan (1991) Linear extensions with backtracking, in: *Adv. Comput. Infor. - ICCI '91* (F. Dehne, F. Fiala, W. W. Koczkodaj, eds.), Springer, 161-168.
- [94] A. Rutkowski (1889) The fixed point property for small sets, *Order* **6**, 1-14.
- [95] M. Saks (1985) Balancing linear extensions of Ordered sets, *Order* **2**, 327-330.
- [96] N. Sauer and E. Woodrow (1984) Finite cutsets and finite antichains, *Order* **1**, 35-46.
- [97] B. Schröder (1992) On the number of nondismantlable posets with the fixed point property, *Order* **8**, 325-329.
- [98] L. N. Shevrin and N. D. Filippov, (1970) Partially ordered sets and their comparability graphs, *Sibirskii Matematicheskii Zhurnal* **11**, 648-677 (translation).
- [99] A. Sidorenko (1981) The Number of Linear Extensions of a Partial Order as a Function of its Incomparability Graph, *Math. Notes* **29**, 40-44.
- [100] A. Sidorenko (1992) Inequalities for the Number of Linear Extensions, *Order* **8**, 331-340.

- [101] R. P. Stanley (1981) Two combinatorial applications of the Aleksandrov-Fenchel inequalities, *J. Comb. Th. Ser. A* 31, 56-65.
- [102] R. P. Stanley (1986) Two Poset Polytopes, *Discrete Comput. Geom.* 1, 9-23.
- [103] G. Steiner (1986) An algorithm to generate the ideals of a partial order, *Oper. Res. Lett.* 5, 317-320.
- [104] G. Steiner (1990) Polynomial algorithms to count linear extensions in certain posets, *Congressus Numeratum* 75, 71-90.
- [105] G. Steiner, On computing the information theoretic bound for sorting: counting the linear extensions of posets, preprint.
- [106] G. Steiner, On counting constrained depth first linear extensions of ordered sets, preprint.
- [107] E. Szpilrajn (1930) Sur l'extension de l'ordre partiel, *Fund. Math.* 16, 386-389.
- [108] W. T. Trotter (1983) Graphs and partially ordered sets, *Graph Theory* 2, 237-268.
- [109] T. Trotter, Algorithmic and constructive properties of linear extensions of ordered sets, preprint.
- [110] W. T. Trotter, W. V. Gehrlein and P. C. Fishburn, Balance theorems for height-2 posets, *Order* 9, 43-53.
- [111] L. G. Valiant (1979) The complexity of computing the permanent, *Theoret. Comput. Sci.* 8, 189-201.

- [112] L. G. Valiant (1979) The complexity of enumeration and reliability problems, *Siam J. Comput.* 8, 410-421.
- [113] R. Wille, Lexicographic Decomposition of ordered sets (graphs), preprint.
- [114] P. Winkler (1982) Average height in a partially ordered set, *Disc. Math.* 39, 337-341.
- [115] P. Winkler (1983) Correlation among partial orders, *SIAM J. Disc. Meth.* 4, 1-7.
- [116] P. Winkler (1985) Random orders, *Order* 1, 317-331.