

A New Order of Things:
The Structure of Partial Orders in the Face of Lower Bounds

A Thesis
Presented to
The Established Interdisciplinary Committee
for
Mathematics and Economics
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Sarah D. Brauner

May 2016

Approved for the Committee
(Mathematics and Economics)

Angélica Osorno

Jon Rork

Acknowledgments

First to my thesis advisors, from whom I have learned an immeasurable amount. To Angélica, for guiding me, patiently, diligently, and forcefully, towards mathematical maturity. For your infectious enthusiasm and valuable insights; for your high standards, your support, and your fierce intellect. For understanding so much, mathematically and otherwise. To Josh, for introducing me not only to the topic of this thesis, but to the world of mathematical research. For your insightful feedback and intuition, for your guidance as I learned how to conduct research, to code, to give talks; for helping put me on the path I am on now. To Jon, for your unquestioning support, enthusiasm, and flexibility; for listening to me as I frantically drew posets on your door.

To the mentors and professors that have made a lasting impact on my life. To Kim, for your brilliance and your strength; for believing in me long before I believed in myself, for fighting to make sure that I learned my own worth. To Adam, for providing me with valuable advice and encouragement when I needed it most. To Jamie, for first opening my eyes to how beautiful math can be.

To my family, for instilling my life with love and support. To my mom, for your strength, your humor and your creativity; for making every electric eel I have ever made light up. To my dad, for your patience, sensitivity and selflessness; for making sure I spent so much of my childhood upside down. To Naomi, for being the person I am closest to genetically, and in every other way. For being so smart, so outspoken, and so funny; for loving me even when I practiced karate on you. To Baba and Zayda, for your unwavering support and devotion; for every Shabbat dinner we celebrated and afternoon we spent together in Bayside. To Doula and Grandad, for your generosity and imaginations; for every pie we baked and state line we traversed together.

To all my friends at Reed. In particular, and in order of their appearance into my life: To Zahra, for your self-possession and incisive wit; for identifying small farmers wherever you see them. To Sasha, for your gaiety, your world-view and the details you fixate on; for Patrick's Next Top Rider and my Wall St. Bald Cap. To Alejandro, for your kindness and boundless enthusiasm, for keeping the flow of soylent constant over the years. To Ezra, for being true to your brand, for abstract and chabad. To Alex, for being my steadfast STEMfemme. To Josh, for your encouragement, levity and steady stream of snapchats; for listening to my thesis talk 3 whole times. To Stella, for the cloth that we were both cut from; for trying to convince me that our theses were secretly the same.

Thank you, I have learned so much from all of you.

*There is nothing more difficult to take in hand,
more perilous to conduct, or more uncertain in its success,
than to take the lead in the introduction of
a new order of things.*

—Niccoló Machiavelli (*The Prince*, 1513)

—Donald Knuth (*Sorting and Searching*, 1973).

Table of Contents

Introduction	1
Chapter 1: Sorting and Partial Orders	3
1.1 Partially Ordered Sets	3
1.1.1 Poset Basics	3
1.1.2 Subposets	5
1.1.3 Linear Extensions	7
1.2 Series-Parallel Posets	8
1.2.1 Composing Posets	8
1.2.2 N-Free Posets	10
1.2.3 The $\frac{2}{3}$ Conjecture for N-Free Posets	12
1.3 Sorting As Preference Revelation	14
1.3.1 Sorting Algorithms	14
1.3.2 Computational Complexity of Sorting	16
Chapter 2: An Introduction to Group Theory	19
2.1 Definition of a Group	19
2.1.1 The Symmetric Group	20
2.1.2 The Order of a Group Element	22
2.2 Subgroups and their Cosets	22
2.2.1 Subgroups	22
2.2.2 Cosets	23
2.2.3 Normal Subgroups	25
2.3 Group Homomorphisms	26
2.3.1 The Natural Map	27
2.3.2 The First Isomorphism Theorem	27
2.4 Group Actions	28
2.4.1 Orbits and Stabilizers	28
2.4.2 Determining the Size of X	29
Chapter 3: Poset Symmetry	31
3.1 The Automorphism Group of a Poset	31
3.1.1 Order Embeddings	31
3.1.2 Order Reversing Maps	32
3.1.3 The Quotient Poset	32

3.2	Linear Extensions and the Automorphism Group	35
3.2.1	The Action of $\text{aut}(P)$ on $\varepsilon(P)$	35
3.2.2	When $e(P) = \text{aut}(P) $	36
3.2.3	When $e(P)$ is prime	38
3.3	Perfect Splits	39
3.3.1	When $ \text{aut}(P) $ is even	40
3.3.2	Self-Dual Posets	41
3.3.3	Serial Parallel Cases	42
Chapter 4: Bound Failure		45
4.1	Introduction to the Problem	45
4.2	Split Failure	46
4.2.1	Small Posets that Fail to Split	46
4.2.2	Split Requirements: Numerics	47
4.2.3	Serial Composition and Splitting	48
4.3	Bound Failure	51
4.3.1	Inheriting Bound Failure	51
4.3.2	The Sequence of Required Splits	52
4.3.3	Bound Failure in the Unordered Set	56
Conclusion		61
Works Cited		63

List of Figures

1.1.6	A Hasse Diagram	5
1.1.12	Comparing y with x or z yields $\chi(P, x, y) = \chi(P, z, y) = \frac{2}{3}$	8
1.2.6	A series-parallel poset	10
1.2.7	An N poset.	10
1.3.3	The questions in a sorting algorithm and its resulting posets.	15
1.3.4	A sorting sequence, $\tau(\{a, b, c\})$	16
2.0.1	A poset with 8 elements.	19
2.1.4	An equilateral triangle.	21
2.3.6	A group homomorphism, where $\hat{\beta}$ is a bijection.	27
3.1.8	A Poset and its Quotient Poset.	34
3.2.5	The first two posets are pointwise decomposable; the last is not.	37
3.2.9	An N poset, which has 5 linear extensions.	39
3.3.5	A Poset P where $\text{aut}(P) \cong \mathbb{Z}/3\mathbb{Z}$	41
3.3.6	Comparing x with y does not yield isomorphic posets.	41
3.3.8	An anti-isomorphic perfect split.	41
3.3.13	A perfect split from a poset with a full N	43
4.2.1	The smallest poset that fails to meet its split.	46
4.2.2	The set of all size 6 posets that fail to meet their split.	47
4.3.4	The ramifications of a $\frac{2}{3}$ split for a poset of size 6.	55
4.3.5	The beginning of the greedy algorithm for $n = 5$	57

Abstract

This thesis explores the structure of posets as it relates to linear extensions, sorting algorithms and the classical information-theoretic lower bound on sorting. In particular, we study the relationship between a poset's automorphism group and its set of linear extensions, as well as the poset of equivalence classes induced by the action of the automorphism group. We prove several new results linking the size and structure of a poset's automorphism group to its set of linear extensions, including a classification of when their orders are equal. We then develop a framework with which to analyze how the information-theoretic bound interacts with the structure of posets, and characterize several sufficient conditions under which posets are structurally compatible with the lower bound. Finally, we evaluate the exactness of the information-theoretic lower bound, and prove several results which suggest that its numerical stipulations are quite incompatible with the structure we associate to partial orders more generally.

Introduction

Individual preferences form the backbone of microeconomic theory, where markets are understood by the interactions of the individuals that comprise them. Each individual behaves in accordance with an intrinsic set of tastes, with the underlying assumption being that participants are necessarily endowed with (or assumed to have) an intuition of how they value certain goods relative to others. It is from this structure at the individual level that we are able to then navigate the territory from single actor to market equilibrium.

This vague notion of an individual possessing an ordered set of tastes can be formalized by designating the goods as ranked members of a set X . We can express a *preference* of good x over good y as $y \leq x$. Traditionally, economic theory has imposed relatively strict conditions on the preference set of actors by requiring them to behave *rationally*, which necessitates that, for a preference set X and goods $x, y, z \in X$:

1. Preferences are complete (trichotomous). That is, for every $x, y \in X$, we have $x \leq y$ or $y \leq x$ (or both).
2. Preferences are anti-symmetric. For $x, y \in X$, if $x \leq y$ and $y \leq x$, then $x = y$.
3. Preferences are transitive. If $x \leq y$ and $y \leq z$, then $x \leq z$ as well.

And yet, in many cases preferences are not fully known, either to the actor themselves or to an authoritative outsider (oftentimes referred to as the benevolent dictator). An individual with preferences that are not completely ranked is said to have *incomplete preferences*, and there is a long tradition of economic analysis that centers around trying to develop a methodology with which to determine a complete preference ordering from incomplete information about an individual. There is a rich literature of the study of preference revelation, which has as its goal to ascertain unknown preference rankings by creating mechanisms through which individuals must make a comparison between two goods; see, for example [14]. Preference revelation thus becomes the process of imposing a binary order relation on two goods, which can then be incorporated into the larger scheme of preferences by using the property of transitivity.

Techniques for preference revelation have been studied in a variety of fields and using a variety of metrics ([3], [8]); what has not been examined, however, is how the structure of preferences evolves if we treat preference revelation as an iterative process of comparisons. In this thesis, we are interested in precisely this; rather than studying the mechanics of *how* a comparison of goods can be implemented, we

focus on how the choice of comparisons given to an individual alters their overall set of preferences, and how the iteration of comparisons interacts with the number of comparisons required to obtain a set of total preferences.

Using the mathematical structure of a *partially ordered set* (poset) to model preferences, we take on the project of investigating these relationships. Information Theory provides a lower bound on the number of comparisons required to obtain a full ordering starting from any given partial ordering, but little is known about how accurate the bound is in predicting best-case scenarios¹, nor how it interacts with the structure of partial orders. In this thesis, we first provide an in-depth study of the structure of partial orderings, and then develop a framework by which to understand how the information-theoretic lower bound interacts with this structure.

In Chapter 1 we introduce partially ordered sets and their relationship to the general sorting problem. We move to the abstract in Chapter 2, offering an introduction to basic concepts in group theory. We then apply this theory to the study of partial orders in Chapter 3. Finally, Chapter 4 focuses upon the mechanisms of the information-theoretic lower bound in sorting, and how this does (and does not) interact with the partial order structure developed earlier.

¹In actuality, the lower bound determines the number of comparisons for the best worst-case scenario, a notion we will formalize in Chapter 1.

Chapter 1

Sorting and Partial Orders

In this chapter, we define a partially ordered set and discuss important terminology and structure attributed to them. We then introduce an important class of posets, the series-parallel poset, and discuss some of its properties. We finish by defining a sorting algorithm through the language of partially ordered sets.

1.1 Partially Ordered Sets

A natural framework with which to understand a set of preferences is with a mathematical object called a *partially ordered set*, which, as its name suggests, is a set in which some (but not necessarily all) pairs of elements in a set have an order relation between them.

1.1.1 Poset Basics

A partially ordered set consists of both a set and order relations over that set. This is often abbreviated to *poset*, and we may use either throughout the remainder of the thesis. Through the lens of preferences, elements in the set are goods, and individual preferences are given by the order relations in the set. Indeed, the properties of a poset, which we give below, match closely the properties that we assume for partial preference orderings:

Definition 1.1.1. A *partially ordered set* (poset) is a set X with a binary relation \leq on X , denoted $P = (X, \leq)$, where P has the following properties:

1. P is reflexive: For any $x \in X$, $x \leq x$.
2. P is antisymmetric: For any $x, y \in X$, if $x \leq y$ and $y \leq x$ then $x = y$.
3. P is transitive: For any $x, y, z \in X$, if $x \leq y$ and $y \leq z$ then $x \leq z$.

For the remainder of this thesis, we will use $x < y$ when $x \leq y$ and $x \neq y$. We will also write P to mean a poset, with the underlying set X and order relation \leq implicit. Our analysis will focus on finite posets, though posets need not be finite.

Importantly, the order relation in a poset is not trichotomous. We say that $x, y \in X$ are *comparable* and write $x \perp y$ if there is an ordering relation between them; in other words if $x < y$ or $y < x$. We say they are *incomparable* and write $x \parallel y$ if neither $x < y$ nor $y < x$, and define $I(x)$ to be the set of elements incomparable to x in P . We call an element with no order relations an *isolated element* or equivalently, a *discrete point*.

One way to describe an element $x \in P$ is by the set of elements which are larger and smaller than it. We call the set of all elements that are less than a particular element x in P the down-set of x , denoted

$$D(x) := \{y \in P : y < x\}. \quad (1.1.2)$$

Analogously, the *up-set* of x is

$$U(x) := \{y \in P : x < y\}. \quad (1.1.3)$$

The *cover* of x is the set of elements that are directly above or below it in an ordering; that is, y covers x (or is a covering of x) if $x < y$ and there is no $z \in P$ such that $x < z < y$. We define the *covering up-set* for $x \in P$ as

$$u(x) = \{y \in P : x < y \text{ and } y \text{ is a covering relation of } x\}$$

and define the *covering down-set*, $d(x)$, analogously.

We say that $x \in P$ is *maximal* if for every $y \in P$, if $y \perp x$, then $y < x$. Equivalently, x is maximal if $U(x) = \emptyset$. Analogously, x is *minimal* if for every $y \in P$ where $y \perp x$, we have $x < y$, which is the same condition as $D(x) = \emptyset$.

Note that the conditions for being maximal and minimal make no requirement of uniqueness. A poset can have multiple maximal and minimal elements—in fact, they often do. We denote the set of minimal elements of P as $\min(P)$ and maximal elements, $\max(P)$. If a poset is finite, we can use the notion of minimality to decompose the poset into levels according to the inductive formula

$$\omega_i = \min(P - (\bigcup_{j < i} \omega_j)), \quad (1.1.4)$$

where $\omega_0 = \min(P)$. This decomposition will prove useful when we analyze the structure of posets more fully in Section 1.2. If a poset P has a unique maximal element, we say that it is the *maximum* of P . Similarly, a unique minimal element of P is its *minimum*.

Hasse Diagrams

Posets can be represented visually by using a *Hasse Diagram*, a graph that depicts order relations within the set. The Hasse Diagram of a poset $P = (X, \leq)$ has as its vertices the elements in X and as its edges the covering relations in P . For $x, y \in P$, we adopt the standard convention that if there is an edge between x and y in the Hasse Diagram and y is drawn above x , then $x < y$. If $x < y$ but y does not cover x , there is a strictly increasing path from x to y in the Hasse Diagram.

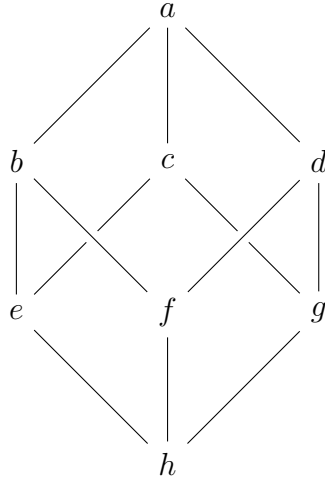


Figure 1.1.6: A Hasse Diagram

Example 1.1.5. Figure 1.1.6 depicts the Hasse Diagram of a poset over the set $X = \{a, b, c, d, e, f, g, h\}$. The elements b, c and d are incomparable, as are e, f and g . Despite the fact that c appears to be above f in the Hasse Diagram, c and f are also incomparable, because there is no strictly increasing path connecting them. The maximum element in P is a , and the minimum element is h . Note that a covers b, c and d , and thus there is an edge between these elements. There is no edge between a and e , for example, despite the fact that $e < a$ because a does not cover e ; rather, we draw edges between the covering relations of e , and infer that $e < a$ through transitivity, by following a path from $e < b < a$.

For the remainder of this thesis, we will regularly use Hasse Diagrams to represent posets.

1.1.2 Subposets

Oftentimes, we are interested in analyzing the restriction of a poset to a *subposet*.

Definition 1.1.7 (Subposet). For $P = (X, \leq)$, we say $Q = (Y, \leq)$ is a *subposet* of P , written $Q \subseteq P$, if $Y \subseteq X$ and for every $x, y \in Y$ if $x < y$ in Q , then $x < y$ in P as well.

For a subposet $Q \subseteq P$ with $x \in Q$, we write $U_Q(x)$ to refer to the upset of x in Q (and analogously for the down-set).

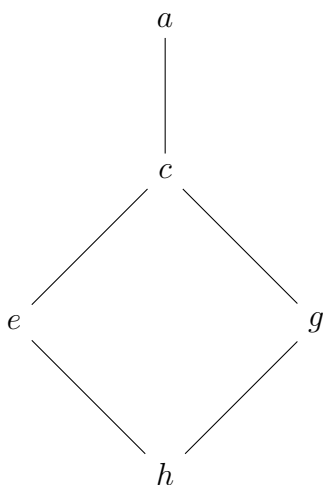
By imposing additional restrictions on subposets, we can require them to retain more of the information of the original poset. The most informative of these restrictions is the *full covering subposet*.

Definition 1.1.8 (Full-Covering Subposet). A poset Q is a *full-covering subposet* of P if

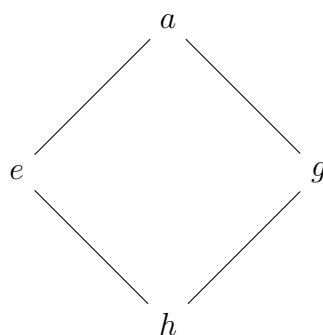
1. Q is a subposet of P .

2. Q is a full subposet: for every x, y in Q , $x < y$ in Q if and only if $x < y$ in P .
3. Q is a full covering: y covers x in Q if and only if y covers x in P .

For example, Figure 1.1.9a shows a full-covering subposet of Figure 1.1.6, while Figure 1.1.9b is a full subposet of Figure 1.1.6 but not covering because it does not include c .



(a) A full-covering subposet of Figure 1.1.6



(b) A full subposet of Figure 1.1.6

For the remainder of this thesis, unless otherwise specified, when we refer to a subposet of P , we will assume it is a full subposet, but not necessarily a covering poset.

The subposets of a poset can tell us a great deal about the structure of the poset as a whole. Of particular interest are the subposets that form a complete order, which we call *chains* and the subposets that are made up entirely of incomparable elements, called *antichains*. We say $Q \subseteq P$ is a *maximal chain* (*maximal anti-chain*) if for any chain (anti-chain) $Q' \subseteq P$, if $Q \subseteq Q'$ then $Q' = Q$.

Like the set of maximal elements of a poset, maximal chains and anti-chains need not be unique. In Figure 1.1.6, for instance, $h < e < b < a$ forms a maximal chain, as does $h < f < d < a$, while b, c, d and e, f, g are both maximal anti-chains. The *height* of a poset is the maximum length of any of its maximal chains, while its *width* is the maximum size of any of its maximal anti-chains. If P is of height 1, we say it is a *discrete poset*.

Chains and anti-chains provide us with an important way to understand a poset, as illustrated by the following theorem, proved by Dilworth in 1950 [4].

Theorem 1.1.10 (Dilworth). *Any poset P over set X with width n can be decomposed into n disjoint chains $X = C_1 \cup C_2 \cup \dots \cup C_n$. Similarly, a poset with height m can be decomposed into m disjoint antichains.*

It is worth noting that there can, and often will be, order relations in P between the elements of the disjoint chains guaranteed by Dilworth's Theorem. Moreover, this decomposition need not be unique (and rarely is.)

1.1.3 Linear Extensions

We are interested not just in the subposets of P , but in what total orderings P can admit. To this end, we introduce the *linear extension*.

A *linear extension* of P is a total order on the underlying set that preserves the existing order relations in P . For a given partial order, P , we denote the set of its linear extensions as $\varepsilon(P)$ and the cardinality as $e(P) = |\varepsilon(P)|$. For a linear extension $L \in \varepsilon(P)$, we write

$$L = (x_1, x_2, \dots, x_n)$$

to mean that L is the total order $x_1 < x_2 < \dots < x_n$, where $x_i \in P$ for every $1 \leq i \leq n$.

The set $\varepsilon(P)$ denotes all possible total orderings compatible with the order relations in P . As such, for any incomparable pair of elements $x, y \in P$, there must be at least one linear extension $L_1 \in \varepsilon(P)$ where $x < y$ in L_1 and some $L_2 \in \varepsilon(P)$ where $y < x$ in L_2 . Furthermore, in $\varepsilon(P)$ the only order relations that vary are those between incomparable elements. It follows that the intersection of the linear extensions is exactly equal to the order relations in P ,

$$P = \bigcap_{L \in \varepsilon(P)} L.$$

Splitting

The properties of linear extensions are closely linked to comparison-based sorting, as the goal of moving from an unordered or partially ordered set to a complete ordering necessitates eliminating all possible linear extensions but one. For an incomparable pair $x||y \in P$, we denote the partial order obtained by comparing x and y as $P_{x<y}$ when $x < y$ (and $P_{y<x}$ if $y < x$). We can generalize this beyond a single comparison: for a poset P with incomparable elements x_1, x_2, \dots, x_n , the poset obtained by learning that $x_1 < x_2 < \dots < x_n$ is denoted $P_{x_1 < x_2 < \dots < x_n}$.

Every time a comparison is made between previously incomparable elements, some number of linear extensions for $\varepsilon(P)$ are necessarily eliminated in the resulting posets. When we compare x to y , for instance, any linear extension L in which $y < x$ cannot be in $\varepsilon(P_{x<y})$. Importantly, L is still a linear extension of the poset $\varepsilon(P_{y<x})$. This tells us that any comparison of incomparable elements $x, y \in P$ partitions $\varepsilon(P)$ by the sets $\varepsilon(P_{x<y})$ and $\varepsilon(P_{y<x})$.

Moreover, learning that $x < y$ may imply other relations through transitivity. Consider, for example, a comparison between c and f in Figure 1.1.6. If $c < f$, then $c < b$ and $c < d$ as well.

We call the proportion of linear extensions which are still feasible after a comparison is made for $x < y$ the *split*, and write $\Upsilon(P_{x<y})$ to be

$$\Upsilon(P_{x<y}) = \frac{e(P_{x<y})}{e(P)}.$$

Naturally, $\Upsilon(P_{x<y}) + \Upsilon(P_{y<x}) = 1$. We designate

$$\max[\Upsilon(P_{x<y}), \Upsilon(P_{y<x})] = \chi(P, x, y).$$

The split of a comparison plays an important role in subsequent chapters, as it partially determines how many comparisons are required to obtain a fully ordered set. For now, we note an important conjecture which provides a lower bound on the best possible split for an arbitrary poset. This is important because it gives us a best-worst case on how evenly we can divide *any* poset, a property that will become highly relevant when we examine how sorting sequences interact with the information-theoretic lower bound in Chapter 3. The Conjecture was first stated by Kislitsyn in 1968 [11] and remains unproven for all posets, although it has been shown to be true for certain classes of posets. Brightwell provides a more comprehensive summary in [2].

Conjecture 1.1.11 (1/3-2/3 Conjecture). *For any poset P , there exists a pair of incomparable elements, $x, y \in P$ such that $\chi(P, x, y) \leq \frac{2}{3}$.*

If true, we know this bound is tight; in the case of the poset given in Figure 1.1.12, a $\frac{2}{3}$ split is the *only* possible split.

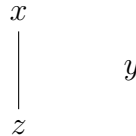


Figure 1.1.12: Comparing y with x or z yields $\chi(P, x, y) = \chi(P, z, y) = \frac{2}{3}$

1.2 Series-Parallel Posets

In this section, we turn to a particularly nice class of poset, the series-parallel poset. Series-parallel posets are constructed entirely from two composition operations. We first discuss these operations, and then some properties that follow from them.

1.2.1 Composing Posets

One method of composing posets is by stacking them on top of each other. We call this a *serial composition*.

Definition 1.2.1. For n disjoint posets $P_1 = (X_1, \leq), P_2 = (X_2, \leq), \dots, P_n = (X_n, \leq)$ we say $P = (X, \leq)$ is a *serial composition* of P_1, P_2, \dots, P_n , written

$$P = P_1 \boxplus P_2 \boxplus \dots \boxplus P_n,$$

if the following conditions hold:

1. The underlying set of P , $X = \bigcup_{k=1}^n X_k$.
2. The order relations in each P_i are preserved. If $x, y \in P_i$, then $x < y$ in P_i if and only if $x < y$ in P .
3. For x in P_i and y in P_j , if $i < j$ then $x < y$ in P .

Note that the order of serial composition is important. The poset $P_1 \boxplus P_2 \neq P_2 \boxplus P_1$; in the latter expression every $x_1 \in P_1$ is greater than every $x_2 \in P_2$, while in the former the opposite is true.

From Definition 1.2.1 it follows that $P = P_1 \boxplus P_2 \boxplus \dots \boxplus P_n$ if and only if every $L \in \varepsilon(P)$ is also a serial composition $L = L_1 \boxplus L_2 \boxplus \dots \boxplus L_n$, where $L_k \in \varepsilon(P_k)$ for $1 \leq k \leq n$.

From this we can conclude that the number of linear extensions of P is given by the product of the linear extensions of its serial components,

$$e(P) = \prod_{k=1}^n e(P_k). \quad (1.2.2)$$

Another useful poset composition operation is the *parallel composition*, which composes posets by taking their disjoint union.

Definition 1.2.3. For n disjoint posets $P_1 = (X_1, \leq), P_2 = (X_2, \leq), \dots, P_n = (X_n, \leq)$ we say $P = (X, \leq)$ is a *parallel composition* of P_1, P_2, \dots, P_n , written

$$P = P_1 \oplus P_2 \oplus \dots \oplus P_n,$$

if the following conditions hold:

1. The underlying set of P , $X = \bigcup_{k=1}^n X_k$.
2. The order relations in each P_i are preserved; $x < y$ in P_i if and only if $x < y$ in P .
3. For every x in P_i and y in P_j , if $i \neq j$ then $x \parallel y$.

In order to determine the linear extensions of a parallel composition between two posets, P_1 and P_2 , we must consider the linear extensions of P_1 and P_2 , as well as the number of possible ways these orderings can be combined. Thus the number of linear extensions of $P_1 \oplus P_2$ is

$$e(P_1 \oplus P_2) = e(P_1) \cdot e(P_2) \cdot \binom{|P_1| + |P_2|}{|P_1|}. \quad (1.2.4)$$

The operations \boxplus and \oplus can be used to construct a poset from smaller posets—but we can also think of them as decomposition operations that break up a larger poset into smaller subposets. As we shall see in subsequent sections, if $Q \subseteq P$ and Q can be obtained by a sequence of \boxplus and \oplus decompositions of P , then certain properties that hold for Q are true for P as well. In this case, we say that $Q \sim P$ *up to series-parallel decomposition*.

An important class of poset comes from the posets that can be decomposed into discrete points using only serial and parallel decompositions. We call this class *series-parallel posets*.

Definition 1.2.5. A poset P with size $|P| = n$ is a *series-parallel poset* if it can be constructed from n discrete points using some sequence of serial (\boxplus) and parallel (\oplus) compositions.

We illustrate this process in Figure 1.2.6, which can be decomposed into the discrete points x, y, z by performing a series decomposition of x and y with z , followed by a parallel decomposition of x and y .

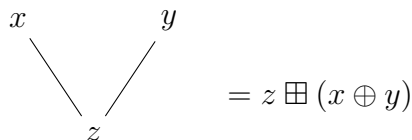


Figure 1.2.6: A series-parallel poset.

We shall see in subsequent sections that this property makes series parallel posets particularly nice to deal with as compared to other classes of posets.

1.2.2 N-Free Posets

Series-parallel posets are characterized by the fact that they can be decomposed into discrete points using only \boxplus and \oplus . We now turn our attention to posets that cannot be decomposed in such a way. In order to do so, we must first introduce some terminology.

Recall that Q is a full subposet of P if $Q \subseteq P$ and for every $x, y \in Q$, $x < y$ in P if and only if $x < y$ in Q .

In this section, the particular full subposet we devote our attention to is the one pictured below, which as its Hasse Diagram shows, has an “N”.

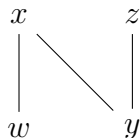
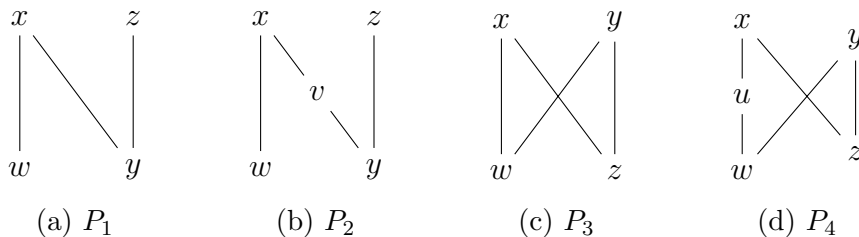


Figure 1.2.7: An N poset.

We define the class of posets that does *not* have an N as follows:

Definition 1.2.8. A poset P is *N-free* if it contains no full subposet, $Q \subseteq P$ where for $w, x, y, z \in Q$, $w < x > y < z$.

To illustrate Definition 1.2.8, consider the posets pictured below.



(a) P_1

(b) P_2

(c) P_3

(d) P_4

1. P_1 is an N -poset, so it must also be a full N subposet of itself.
2. P_2 has a full subposet N : $w < x > y < z$.
3. P_3 has a subposet N : $w < x > y < z$ but it is not full because we must omit $w < y$ to make an N .
4. P_4 has two full N subposets: $u < x > z < y$ and $u > w < y > z$.

One important characteristic of N subposets is that the elements that form the N do not share the same up-set or down-set. This is closely linked to serial and parallel composition; Rival [19] provides a proof that any poset which contains a full N subposet cannot be decomposed fully using serial and parallel compositions. We now present that proof.

Theorem 1.2.10 (Rival). *A poset P is series-parallel if and only if it is N -free.*

Proof. Let P be series-parallel. Then it can be decomposed into discrete points using \boxplus and \oplus operations. Note that by the definition of a full subposet, if $Q \subseteq P$ cannot be decomposed using these operations, then P cannot either. No full N subposet can be decomposed using \boxplus and \oplus , so P must be N -free.

Conversely, let P be a finite N -free poset. We prove by induction on $|P|$ that P can be decomposed into discrete points using \boxplus and \oplus , and is therefore series-parallel. The base case is straightforward: for any poset P where $|P| \leq 3$, P can easily be decomposed using \boxplus and \oplus .

Assume that for any N -free poset Q , $|Q| < |P|$, Q is series-parallel. If P has a unique minimum or maximum element, x , then we can perform a serial decomposition, $P = \{x\} \boxplus P - \{x\}$, and so we are done. Similarly, if P is not connected, then its disjoint components form a parallel composition. Thus without loss of generality, let P be connected and have no unique minimum or maximum element.

We first show that every maximal element is greater than every minimal element in P . Let x be maximal and y be minimal in P . Then because P is connected, there is some sequence of elements $x = z_0 > z_1 < z_2 > z_3 < \dots > z_k = y$ in P . Choose the shortest such sequence. By our assumption, k must be 1 to avoid creating a full N . So $x > y$.

Now, define M to be the set consisting of every minimal element of P . Consider the intersection of the up-sets of the elements of M ,

$$I = \bigcap_{m \in M} U(m).$$

Note that every maximal element of P is in I by the above argument, and by assumption $I \neq \emptyset$. Furthermore, $I \cap M = \emptyset$ because $|M| \neq 1$ by assumption. Similarly $P - I \neq \emptyset$.

We now show that $P = (P - I) \boxplus I$ to complete the proof. Suppose, for the sake of contradiction, that for $x \in I$ and $y \in P - I$, $x \not\prec y$. Then $y \notin M$ by the definition of I . Thus there must be some $m \in M$ such that $m < y$. Because $x \in I$, $m < x$ as well. Since $y \notin I$, there is another element $m' \in M$ where $m' \not\prec y$ but $m' < x$. But then we have $m' < x > m < y$ which forms a full N .

So $x > y$ for every $x \in I$, $y \in P - I$, which is precisely the condition for which $P = (P - I) \boxplus I$.

□

1.2.3 The $\frac{2}{3}$ Conjecture for N-Free Posets

Theorem 1.2.10 adds rigor to our intuition that series parallel posets are a “nice” class of poset, and ensures that any N-free poset has a simple formula with which to count linear extensions. Zaguia [24] proves that series-parallel posets also split nicely; that is, N-free posets always have a pair of elements, x, y such that $\chi(P, x, y) \leq \frac{2}{3}$. We present the theorem and proof below.

Theorem 1.2.11 (Zaguia). *Let P be an N-free poset. Then there exist a pair of incomparable elements $x, y \in P$ such that $\chi(P, x, y) \leq \frac{2}{3}$.*

We prove several lemmas for N-free posets first.

Lemma 1.2.12. *Let P be an N-free poset. Then for $x, w \in P$ if $u(x) \cap u(w) \neq \emptyset$ then $u(x) = u(w)$, and analogously for $d(x)$ and $d(w)$.*

Proof. Suppose not. Then there is some $z \in u(x) \cap u(w)$ and $y \in u(x)$, $y \notin u(w)$. If $y||w$, then we have that $w < z > x < y$, which forms a full N . So assume $y > w$. Since $y \notin u(w)$, there must be some $t \in u(w)$ such that $y > t > w$. Because $y \in u(x)$, $x||t$. But then we have $t > w < z > x$, which is also a full N subposet. Thus there can be no such y . □

Recall by Equation 1.1.4 that we can decompose any finite poset by its set of minimal elements, such that $\omega_0 = \min(P)$ and $\omega_i = \min(P - (\bigcup_{j < i} \omega_j))$.

Lemma 1.2.13. *Let P be an N-free poset with height h . Then for every $x \in P$, there exists an $i \leq h$ such that $u(x) \subseteq \omega_i$.*

Proof. Suppose $x \in P$ has distinct upper covers y and z belonging to distinct levels $y \in \omega_j$ and $z \in \omega_k$, with $j < k$. Then z has a lower cover $w \in \omega_{k-1}$. Because y and z are in $u(x)$, we have $y||z$. Similarly, because $x, w \in d(z)$, $x||w$.

We also have that $y||w$. To see why, note that $j \leq k - 1$, so $w \not\prec y$. If $y < w$, then $y < w < z$, so $z \notin u(x)$. Thus $y||w$.

This leaves us with $w < z > x < y$, which is an N and therefore contradicts the hypothesis. So x cannot have upper covers in distinct levels.

□

Lemma 1.2.14. *Let i be the largest value such that there are two distinct elements $x_1, x_2 \in \omega_i$ where $d(x_1) = d(x_2)$. Then for every $x \in \omega_i$, $U(x) \cup \{x\}$ is a chain.*

Proof. Let $x \in \omega_i$ and suppose $U(x)$ is not a chain. Without loss of generality, $U(x)$ is not empty. Then there is some $y \in U(x) \cup \{x\}$ with distinct upper covers z, w . From Lemma 1.2.13 z and w must be in the same ω_k , $k > i$. But then $d(z) = d(w)$, contradicting the choice of i . \square

Lemma 1.2.15 is true for any poset.

Lemma 1.2.15. *Let P be a poset. Suppose there is a pair of incomparable elements $x, y \in P$ where $U(y) \subseteq U(x)$ and $D(x) \subseteq D(y)$. Then $\Upsilon(P_{x<y}) \geq \frac{1}{2}$.*

Proof. Consider $\varepsilon(P_{x<y})$ and $\varepsilon(P_{y<x})$. Suppose there is an $L \in \varepsilon(P_{y<x})$ such that $y < z < x$ in L for some $z \in P$. Then z must be incomparable to x and y in P , as $U(y) \subseteq U(x)$ and $D(x) \subseteq D(y)$. Thus swapping x and y and fixing all other elements yields some $L' \in \varepsilon(P_{x<y})$. We define a map $\psi : \varepsilon(P_{y<x}) \rightarrow \varepsilon(P_{x<y})$ that swaps x and y and fixes all other elements in the order. It follows that ψ is injective because we only move x and y . Thus $e(P_{y<x}) \leq e(P_{x<y})$, so $\Upsilon(P_{x<y}) \geq \frac{1}{2}$. \square

At last, we can prove Theorem 1.2.11.

Proof of Theorem 1.2.11. Let P be N -free and let $\omega_0, \dots, \omega_h$ be the sequence of levels given by Equation 1.1.4. If P has a unique minimum, it must be the smallest element in any linear extension, and so has no effect on $\chi(P, x, y)$. Thus, without loss of generality, let $|\omega_0| \geq 2$.

Choose i , $0 \leq i \leq h$ such that i is the largest value in which ω_i has two elements x, y with $d(x) = d(y)$. If $U(x) = U(y) = \emptyset$, then $\Upsilon(P_{x<y}) = \frac{1}{2}$ and the proof is complete, so assume this is not the case.

Without loss of generality, let $U(y) \neq \emptyset$. By Lemma 1.2.14, $U(y) \cup \{y\}$ is an n -chain $y = y_1 < y_2 < \dots < y_n$.

For the sake of contradiction, suppose that $\Upsilon(P_{x<y}) < \frac{1}{3}$. We define a probability distribution with the following events:

$$q_1 = \Upsilon(P_{x<y_1}),$$

$$q_j = \Upsilon(P_{y_{j-1}<x<y_j}) \text{ for } 2 \leq j \leq n \text{ and}$$

$$q_{n+1} = \Upsilon(P_{y_n<x}).$$

Naturally, $\sum_{j=1}^{n+1} q_j = 1$ because the events are mutually exclusive and together make up $\varepsilon(P)$.

We now show that $0 \leq q_{n+1} \leq \dots \leq q_1 < \frac{1}{3}$. To this end, introduce the mapping

$$\phi : \varepsilon(P_{y_j<x<y_{j+1}}) \rightarrow \varepsilon(P_{y_{j-1}<x<y_j})$$

such that for $L \in \varepsilon(P_{y_j<x<y_{j+1}})$, ϕ swaps x and y_j in the ordering and fixes all other elements.

Claim 1. *The mapping ϕ is well defined.*

Take any $L \in \varepsilon(P_{y_j < x < y_{j+1}})$. Clearly $y_j || x$ in P , so if y_j covers x in L , then $\phi(L) \in \varepsilon(P_{y_{j-1} < x < y_j})$. Thus suppose that there is some $z \in P$ such that $y_{j-1} < z < x$ in L . Because $y_1 < \dots < y_n$ forms a chain, if $y_{j-1} < z$ in P , then $y_{j+1} < z$ in P as well. Because $z < x$ in L , $z < y_{j+1}$ in L , so $z || y_{j-1}$ in P . We also have that $z || x$ in P because $D(x) = D(y_1)$ by assumption. Therefore $\phi(L) \in \varepsilon(P_{y_j < x < y_{j+1}})$.

Claim 2. *The mapping ϕ is injective.*

This follows immediately because we move only x and y_j .

Thus $0 \leq q_{n+1} \leq \dots \leq q_1$. By assumption, $q_1 < \frac{1}{3}$.

Now, choose k so that it satisfies:

$$\sum_{j=1}^{k-1} q_j \leq \frac{1}{2} < \sum_{j=1}^k q_j.$$

As $\sum_{j=1}^{k-1} q_j = \Upsilon(P_{x < y_{k-1}}) \leq \frac{1}{2}$, it must be less than $\frac{1}{3}$ or we are done; so $\sum_{j=1}^{k-1} q_j < \frac{1}{3}$. Analogously, because $\frac{1}{2} < \sum_{j=1}^k q_j$, it follows that $\sum_{j=1}^k q_j > \frac{2}{3}$. So $q_k > \frac{1}{3}$, but we've already shown that $\frac{1}{3} > q_1 \geq q_k$. Thus $q_1 \geq \frac{1}{3}$, and so $\chi(P, x, y) \leq \frac{2}{3}$. \square

1.3 Sorting As Preference Revelation

1.3.1 Sorting Algorithms

A partial order on a set offers a mathematical structure with which to represent a static preference configuration—but we are also interested in dynamic sets of preferences. That is, we would like to study the structure of posets as order is revealed through comparison. One way to formulate this is as the process by which we obtain a complete ordering from a partial order, which we do by making a sequence of comparisons between incomparable elements. This is precisely a comparison-based sorting algorithm.

As its name suggests, comparison-based sorting algorithms make comparisons between the elements they are sorting, and use the information gained from that answer to inform subsequent comparisons. This means two things; first, that such an algorithm must store the order relations that it knows, and second, that after each comparison we must have two separate sets of information. In order to ensure that *any* input can be sorted, we must consider both sets of answers to a comparison, and take the “worse” of the two—in other words, the answer that provides less information.

This should sound quite familiar given the concepts discussed earlier in this chapter, which provide us with the knowledge necessary to formalize such a process: the partial information being stored and used to determine subsequent comparisons is perfectly described by a poset. Given a poset P with $x, y \in P$, we can make a comparison between x and y , which yields two answers: $P_{x < y}$ and $P_{y < x}$. We thus have all the components of a sorting algorithm, and merely need to put them together.

We do so using the concept of a *binary decision tree*, a data structure in which each node, beginning with a *root node*, has two children. In our tree, each node will

store a comparison between a pair of elements, and each child will be the resultant poset from that comparison. When we arrive at a total order the algorithm stops, and we call the last node, which has no children, a *leaf*. We can thus define a *sorting algorithm* to be the decision tree with a root determined by the partial information that we begin with—we do not stipulate that our sorting algorithm have as input an unordered set.

Of course, we are not just interested in the sequence of comparisons required to obtain a total order, but also the posets that accompany those comparisons, and their relationships to one another. For this reason, we define a sequence, τ , to accompany any sorting algorithm, where τ is made up of the posets corresponding to each step (comparison) in a sorting algorithm.

Definition 1.3.1. A *sorting sequence*, τ over a set X is the sequence of partial orders, $\tau(X) = P_0, P_1, \dots, P_j$, obtained by traversing a path in a sorting decision tree. We assume that P_j is a total order, but do not require that P_0 be a discrete poset.

These definitions allows us to explore the entirety of the space of sorting algorithms rather than restricting our study to optimal algorithms. We do so for several reasons: if we wish to gain an understanding of structural patterns across algorithms, we must study the space as a whole. Moreover, in some cases, we may be interested precisely in the structural relationships between suboptimal algorithms and their posets. Note, also, that this framework generalizes a sorting algorithm so that we are able to define the first term P_0 as an arbitrary poset rather than with a completely unsorted list.

We provide the following example of a sorting algorithm for the set $\{a, b, c\}$.

Example 1.3.2. We sort $\{a, b, c\}$ by making the comparisons

$$\{(a \stackrel{?}{<} b), (b \stackrel{?}{<} c), (a \stackrel{?}{<} c)\},$$

which we can depict pictorially as in Figure 1.3.3. We have not shown the whole decision tree, but rather given the answers for each comparison. We contrast this with Figure 1.3.4, where we show a sorting sequence, $\tau(\{a, b, c\})$.

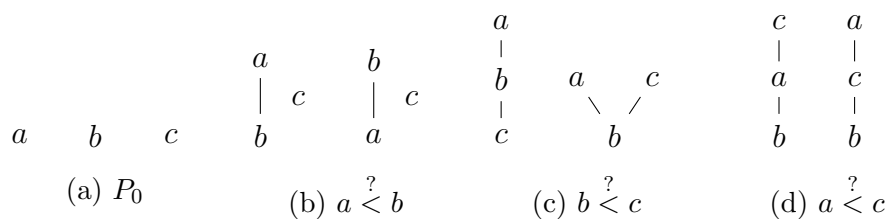
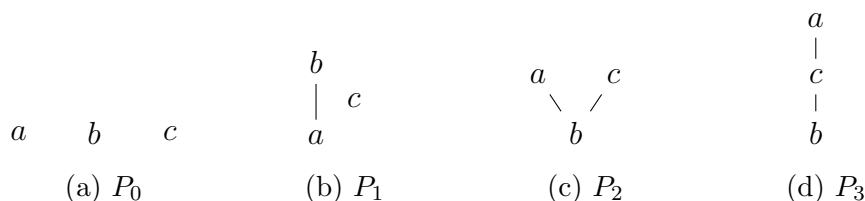


Figure 1.3.3: The questions in a sorting algorithm for $\{a, b, c\}$ and its resulting posets.

Figure 1.3.4 differs from Figure 1.3.3 because we choose one answer to each question rather than both. This is precisely what it means to traverse a path in the decision tree. Note that when the linear extension split from a comparison is uneven, we chose the poset with the larger number of linear extensions, and when the split is perfectly even (as in the comparison $a \stackrel{?}{<} b$), we chose a poset arbitrarily.

Figure 1.3.4: A sorting sequence, $\tau(\{a, b, c\})$

1.3.2 Computational Complexity of Sorting

In the computational model we have introduced, we characterize complexity by the number of comparisons required to obtain a total order, which can also be understood as the length of a path in the decision tree. Sorting is relatively unique in that asymptotically, its upper and lower bounds can be bounded by $n \log n$; that is, there exist constants c_1, c_2 such that for sufficiently large values of n , the computational time for a given sorting algorithm, $f(n)$ can be bounded by

$$c_1 n \log(n) \leq f(n) \leq c_2 n \log(n).$$

The upper bound of $c_2 n \log(n)$ tells us that for any n there exists a decision tree whose depth is at most $c_2 n \log(n)$, while the lower bound of $c_1 n \log(n)$ tells us that every path in a decision tree must have at least $c_1 n \log(n)$ steps. However, in this thesis, we are primarily interested in the lower bound, and in particular, in its exactness rather than its asymptotic behavior.

A Precise Lower Bound

For the lower bound of n , we can be quite specific about the minimum number of questions we must ask to obtain a full sorting.

We begin with the case for an unsorted list, X , where $|X| = n$, and therefore $e(X) = n!$. We aim to make comparisons until we are left with a single linear extension. Because we take the worse split, in the best case, for every pair of incomparable elements $x, y \in X$, $\chi(P, x, y) = \frac{1}{2}$. This leaves us with $\log_2 n!$ comparisons.¹ Of course, in order to use this bound in sorting, we round up to the nearest integer, giving us:

$$|\tau(X)| \geq \lceil \log_2(n!) \rceil.$$

This describes the *information-theoretic lower bound* for an unsorted list, but we can expand the applicability of this lower bound to any partial order using the same logic: we aim to take the set of linear extensions for P and eliminate all but one. We refer to the number of comparisons stipulated by the information-theoretic lower bound for a given P as $\mathbb{B}(P)$. Each question eliminates at best half of the linear extensions, and so we have:

¹By Stirling's approximation, $\log_2 n! \approx n \log n$, so asymptotically, we make approximately $n \log_2(n)$ comparisons to obtain a fully sorted list. This is where the $c_1 n \log(n)$ comes from.

$$|\tau(P)| \geq \mathbb{B}(P) = \lceil \log_2 e(P) \rceil.$$

The exact tightness of this bound is still an open question, and a central motivation for this thesis. It has been known for at least 50 years that for a set of size n , for $12 \leq n \leq 19, n = 22$ the minimum number of comparisons needed is one more than the information-theoretic bound, while for $n \leq 11, n = 20, 21$, the bound is tight [22], [23]. Moreover, for $n > 23$, we have no conception of the tightness of the information-theoretic bound. We refer to a set as *bound-failing* if there exists no sorting algorithm that can be completed using only the number of comparisons stipulated by the information-theoretic lower bound. The fact that our knowledge of bound-failure for discrete posets comes from exhaustive computer search indicates the lack of mathematical understanding we have of what determines bound failure. This is of direct relevance to the questions raised earlier, in which we are concerned with the underlying structure of sorting as preference revelation.

Thus if we wish to investigate how partial preferences interact with sorting, we need to develop a mathematical understanding of the structure of partially ordered sets and their symmetries—and for this, we require a knowledge of group theory.

Chapter 2

An Introduction to Group Theory

This chapter presents an introduction to the theory of groups, which we will use in subsequent chapters (particularly Chapter 3) to analyze in greater detail the structures of posets. We begin with an example.

Looking at the poset in Figure 2.0.1, we can use what we have learned in Chapter 1

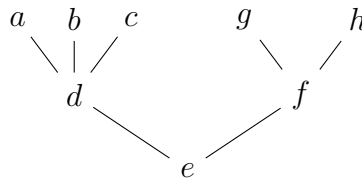


Figure 2.0.1: A poset with 8 elements.

to determine the up-sets and down-sets of each element (among other things). What is less clear is whether any of these elements are, in some sense, “equivalent”. In particular, we are interested in equivalence as it pertains to preserving the pre-existing order relations in our poset. Are there any elements that we can move around which leave the poset fundamentally unchanged?

To answer this question, note first that swapping elements a and b does nothing to the order relations in our poset, nor does swapping b with c or a with c . In fact, the structure of our poset is such that no matter how we permute a , b and c , amongst each other, we change no order relations. The same is true of elements g and h , but is not of d and f . Swapping d with f alone would necessarily alter the poset, even though $d \parallel f$, as $|U(d)| = 3$ while $|U(f)| = 2$. We can see then, that incomparability does not imply equivalence of elements, but equivalence of distinct elements does necessitate incomparability. In order to formalize this notion of equivalence, we must first introduce the concepts of groups, and eventually, group actions.

2.1 Definition of a Group

A group is an algebraic structure consisting of a set of elements and an operation that combines those elements, subject to certain axioms.

Definition 2.1.1. A *group* $(G, *)$ is a set G , with a closed binary operation $*$ such that the following hold:

1. G has an identity element: there is an $1_G \in G$ such that $1_G * x = x * 1_G = x$ for every $x \in G$.
2. Each element of G has an inverse: for every $x \in G$ there exists an element $x^{-1} \in G$ such that $x * x^{-1} = x^{-1} * x = 1_G$.
3. $*$ is associative: $(x * y) * z = x * (y * z)$ for all $x, y, z \in G$.

Importantly, groups need not be commutative; a group that is commutative is called *abelian*. Oftentimes, the group operation is multiplication, and so the notation $x * y$ is simplified to xy , a convention which we follow for the remainder of the thesis.

2.1.1 The Symmetric Group

One of the most important groups is the group of symmetries of a set, called the *symmetric group* on a set. For any set X , there is a symmetric group of X , which we write as $\text{Sym}(X)$.

Definition 2.1.2. For a set X , the *symmetric group of X* is

$$\text{Sym}(X) := \{\pi : X \rightarrow X : \pi \text{ is a bijection}\}.$$

The elements that make up $\text{Sym}(X)$ are *permutations* of elements in the set, where a permutation is a function from the set to itself which rearranges elements in that set. Any permutation can be written as a composition of *cycles*, and each cycle describes how elements are moved, while the length of a cycle is equal to the number of elements moved. A *k-cycle* moves k elements. A k -cycle of elements x_1, x_2, \dots, x_k in a set X is written

$$(x_1 x_2 \cdots x_k),$$

which indicates that $x_1 \mapsto x_2$, $x_2 \mapsto x_3$, and so on until the last element, x_k is sent back to the first element, x_1 , thereby completing the cycle. For example, if we mark elements by indices 1, 2, 3, 4 we can denote the permutation that sends $1 \mapsto 2$ and $2 \mapsto 1$ as $(12)(3)(4)$, where (3) and (4) tell us that these indices are fixed. Oftentimes we shorten this to (12) , where $3 \mapsto 3$ and $4 \mapsto 4$ is implicit.

We can take the product of permutations by composing them; when we do so, we perform the permutations from right to left, in the same way that composed functions are evaluated. This convention follows from the view of permutations as functions, and multiplying permutations as a special case of function composition. For example $(12)(23)$ sends:

$$1 \mapsto 1 \mapsto 2$$

$$2 \mapsto 3 \mapsto 3$$

$$3 \mapsto 2 \mapsto 1.$$

Which in total gives us $(12)(23) = (123)$. This operation is not commutative: $(12)(23) \neq (23)(12)$. Importantly, any permutation can be written uniquely (up to order) as the product of disjoint cycles, and disjoint permutation cycles commute.

With this knowledge of permutations, we can turn to a simple example of the symmetric group.

Example 2.1.3 (S_3). We first focus on a set of size 3, $X = \{1, 2, 3\}$. The symmetries of X are all the ways that we can permute the elements 1, 2, 3. Note that we can depict this set as the equilateral triangle shown in Figure 2.1.4, where each vertex is labeled with an element in X , and we permute elements in X by permuting the vertex label of the triangle. How many different ways can we permute the vertices?

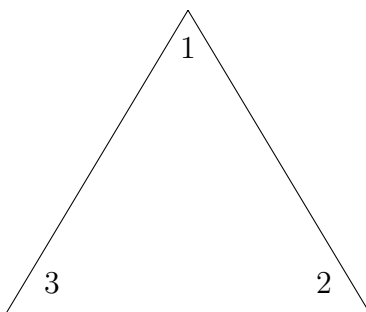


Figure 2.1.4: An equilateral triangle.

There is one permutation that swaps 1 and 2, (12) . Likewise, we can swap 2 and 3, (23) , and 1 and 3, (13) . We can also rotate every element clockwise by 60° : (123) , and similarly we can send every element counter-clockwise by 60° , giving us (132) . Finally, we can “permute” by doing nothing: $(1)(2)(3)$, which we shall write as (1) .

These six permutations form the group S_3 , the *symmetric group of size 3*:

$$S_3 = \{(1), (12), (13), (23), (123), (132)\}.$$

We can check that S_3 obeys the group axioms as follows.

1. There is an identity permutation, (1) , where we don't swap anything.
2. Every permutation has an inverse; applying (12) , (13) or (23) twice brings us back to where we started, while multiplying $(123)(132) = (132)(123) = (1)$
3. Multiplying the permutations is associative. This follows from the fact that permutations are functions.

It is no coincidence that the cardinality of S_3 , $|S_3| = 3! = 6$. In fact, for a set $\{1, 2, \dots, n\}$, we can form the symmetric group of n elements, S_n , where $|S_n| = n!$.

The symmetric group is extremely important in the study of groups generally, and will play a large role in our study of posets as well.

2.1.2 The Order of a Group Element

In any group, we can talk about the *order* of the elements in the group.¹ For a group G we say $g \in G$ is of order k if k is the smallest positive integer such that $g^k = 1_G$. We denote this as $|g| = k$. For example, in S_3 , the order of (132) is 3, because $(132)(132)(132) = (1)$. The order of (12) is 2, because applying (12) twice brings us back to the identity element (1) . When a permutation moves only 2 elements, we say that it is a *transposition* of elements; for instance, (12) *transposes* 1 and 2. Of course, for an infinite group, we may have elements of infinite order, which is to say, there is no k such that $g^k = 1_G$.

The orders of elements in a group tell us much about the structure of the group. Somewhat confusingly, the order of a group G , also written $|G|$, is the cardinality of the group. We can relate the order of a finite group with the order of its elements in the following theorem, which will prove useful in Chapter 3.

Theorem 2.1.5 (Cauchy.). *Let G be a finite group and let p be a prime number dividing $|G|$. Then there is some $g \in G$ such that $|g| = p$.*

2.2 Subgroups and their Cosets

We can learn a lot about the structure of a group by looking at its subgroups and corresponding set of cosets, which we define and discuss below.

2.2.1 Subgroups

Just as we can form a subposet from the restriction of a poset, we can form a subgroup from the restriction of a group. A subgroup of any group inherits the group's operation, and, in order to preserve group structure, any subgroup must possess the same group properties enumerated in Definition 2.1.1. We ensure this with the following conditions:

Definition 2.2.1. A subset $H \subseteq G$ is a *subgroup* of a group G if:

1. H contains the identity element of G , 1_G
2. Every element of H has an inverse also in H ; for every $h \in H$, $h^{-1} \in H$ as well.
3. H is closed under the operation of G . That is, if $x, y \in H$ then $xy \in H$.

In this case, we write $H \leq G$.

No matter the group, we can always form a subgroup from the identity element, which we call the *trivial subgroup*. Moreover, a group is always a subgroup of itself.

¹Not to be confused with the order relation \leq in a poset!

Example 2.2.2. Returning to the poset in Figure 2.0.1, suppose we are interested primarily in the permutations that transpose g and h . We can form a *subgroup* of G , H that involves such permutations, by including 1_G , the permutation that maps every element to itself, and the permutation (gh) . This fulfills the required properties; no matter how many times we compose (gh) and 1_G , their combination will always amount to swapping h and g or leaving them untouched. Moreover, if we apply (gh) twice, we always return to where we started, so every permutation in our subgroup can be reversed. Thus $H = \{1_G, (gh)\}$ is a subgroup of G .

2.2.2 Cosets

The *cosets* of a group allow us to understand how subgroups exist within the greater group structure. Just as we can multiply² elements in a group, we can multiply elements by a subgroup by multiplying a chosen group element, g , with each member of the subgroup. We denote this set as gH , and call it a *coset* of H in G . Cosets are a rather important concept in group theory, as they allows us to partition the group into subsets of equal size.

Definition 2.2.3 (Cosets). For a group G with a subgroup H , we define a *left coset* of H in G with respect to $g \in G$ to be the set

$$gH = \{gh : h \in H\},$$

and the right coset of H in G with respect to g is the set

$$Hg = \{hg : h \in H\}.$$

We denote the set of left cosets of H in G by

$$G/H = \{gH : g \in G\},$$

and the set of right cosets

$$H \backslash G = \{Hg : g \in G\}.$$

If G is abelian, $G/H = H \backslash G$, and for any group we can form a bijection that maps $G/H \rightarrow H \backslash G$. Thus, for the remainder of this thesis, we refer to the left cosets, with all properties holding analogously for right cosets.

Intuitively, the cosets of a subgroup allow us to understand how the subgroup interacts with the rest of the group. Two cosets xH and yH are equal if and only if $xy^{-1} \in H$, which is equivalent to saying that $xH = yH$ if and only if there is some $h \in H$ for which $xh = y$. If there is no such h , there can be no common elements in the two cosets, and they will be disjoint. From this it follows that cosets are either disjoint or identical. Because we compose every element of G with H , every element of the group must appear in some coset. Moreover, when we form gH , we multiply g with every element in H . Thus, the size of every coset of H is $|H|$, so the cosets of

²For linguistic ease, we use multiply here to mean the group operation, which need not be multiplication.

a subgroup are always the same size. From this we conclude an important property of groups: they are partitioned by their cosets. In other words, every element $g \in G$ belongs to exactly one coset of G with respect to an any subgroup H .

A significant consequence of the fact that cosets partition a group is that we can relate the order of G to the number of cosets it has under H . We define the *index* of H in G as the cardinality of G/H , written $|G/H| = |G : H|$. Note that $|G : H| = 1$ only when $H = G$, so that every $g \in G$ is a member of H as well. Lagrange provides us with far-reaching result about the precise relationship between $|G|$ and $|G : H|$.

Theorem 2.2.4 (Lagrange). *Let G be a group and $H \leq G$. Then*

$$|G| = |G : H| \cdot |H|. \quad (2.2.5)$$

Note that Theorem 2.2.4 means that the order of any subgroup always divides the order of the group. We do not restrict our study of groups to finite groups, and so if G is infinite, then Theorem 2.2.4 tells us that $|G : H| \cdot |H|$ must also be infinite.

Example 2.2.6. Let's return to our poset in Figure 2.0.1 and the subgroup $H = \{1_G, (gh)\}$. What are its cosets?

To answer this, we must first determine the permutations that make up G . We can permute between a, b and c in $3!$ different ways and between g and h in $2!$ ways. Any element that permutes a, b, c could simultaneously permute g and h , and thus in total we have $3! \cdot 2! = 12$ possible permutations. Of course, this is only part of our story; there are multiple groups of order 12, and thus we need more information to understand the structure of G .

Because $|H| = 2$, we know by Theorem 2.2.4 that $|G : H| = 6$; thus, G will be partitioned into 6 cosets. We can determine these cosets by multiplying each of the elements in G with H .

Take, for example, the permutation $(ab)(gh)$. We have that

$$((ab)(gh))1_G = (ab)(gh)$$

and

$$((ab)(gh))(gh) = (ab).$$

If we take the permutation (ab) , we similarly get that

$$(ab)1_G = (ab)$$

and

$$(ab)(gh) = (ab)(gh).$$

These permutations form a single coset: $\{(ab)(gh), (ab)\}$, which we can write as $(ab)H$.

This tells us a lot about G/H ; because H does not move a, b or c , the only way that the composition of some $\pi \in G$ with an element in H can permute these elements is if π itself does. Moreover, composing π with H always generates one permutation which swaps h and g and one which fixes them.

By a similar process, we partition our group of 12 elements into 6 cosets:

$$G/H = \{H, (ab)H, (bc)H, (ac)H, (abc)H, (acb)H\}.$$

Note that these cosets partition G based upon the permutations of a, b and c , and that one coset is the subgroup itself, which is made up precisely of the elements that do not move a, b and c . In fact, G/H is also a group—a fact which we will explain and formalize shortly. In order to do so, we first introduce a particular class of subgroup: the normal subgroup.

2.2.3 Normal Subgroups

The operation of a group need not be commutative, and thus we pay special attention to the subgroups for which right and left cosets are equal. We call these *normal subgroups*.

Definition 2.2.7 (Normal Subgroups). A subgroup N is *normal* to G , written $N \trianglelefteq G$, if for every $g \in G$;

$$gN = Ng.$$

It is routine to check that this statement is equivalent to

$$G/N = N \backslash G.$$

The importance of normal subgroups comes from their cosets: it turns out that the condition of normalcy, as defined above, is precisely the condition needed to ensure that G/N is a group with the same operation as G . We call this group the *quotient group*, and it plays a fundamental role in group theory.

Example 2.2.8. Consider the group of integers, \mathbb{Z} , with the group operation as addition. $G = (\mathbb{Z}, +)$ is abelian, so every subgroup is normal. We form a subgroup of the even integers, $2\mathbb{Z}$. To check that this is a subgroup, we note that the identity element, 0 is in $2\mathbb{Z}$, and that for every $z \in 2\mathbb{Z}$, we also have $-z \in 2\mathbb{Z}$. Finally, the sum of two even numbers is even, so $2\mathbb{Z}$ is closed under the operation.

G has two cosets with respect to $2\mathbb{Z}$; the even integers and the odd integers. The corresponding quotient group is thus made up of two elements: $2\mathbb{Z}$, which contains the even integers, and $1 + 2\mathbb{Z}$, the odd integers. So

$$\mathbb{Z}/2\mathbb{Z} = \{2\mathbb{Z}, 1 + 2\mathbb{Z}\},$$

which can be understood as the remainder when we divide any integer by 2. This is addition modulo 2, and we often write $\mathbb{Z}/2\mathbb{Z}$ to signify just that! We can generalize this concept to addition modulo n for any $n \in \mathbb{N}$ by taking the subgroup $n\mathbb{Z}$, and forming the quotient group $\mathbb{Z}/n\mathbb{Z}$.

To fully understand the importance of the quotient group, we must adjust our focus slightly. We move now from the structure of a single group to mappings *between* groups.

2.3 Group Homomorphisms

Groups have a rich structure, and it is important that we be able to move between them in a way that does not destroy this structure. We can ensure that our group structure is preserved by defining a class of mappings called *group homomorphisms*.

Definition 2.3.1. For groups G and K , we say $\gamma : G \rightarrow K$ is a *homomorphism* if for every $x, y \in G$,

$$\gamma(xy) = \gamma(x)\gamma(y).$$

The *image* of γ is

$$\text{Im}(\gamma) = \{\gamma(x) : x \in G\}$$

and the *kernel* of γ is

$$\ker(\gamma) = \{y \in G : \gamma(y) = 1_K\}.$$

In the case where $\text{Im}(\gamma) = K$ and $\ker(\gamma) = 1_G$, we say that γ is an *isomorphism*, and thus a bijection between groups. When two groups are isomorphic, we write $K \cong G$, and we tend to think of two structures which are isomorphic to each other to be, in some senses, “the same.” To give intuition to this important idea, we provide two examples.

Example 2.3.2. Consider again the subgroup $H = \{1_G, (gh)\}$. If we apply (gh) twice, we get 1_G . This subgroup is actually a disguised version of $\mathbb{Z}/2\mathbb{Z}$! We can define an isomorphism $\beta : H \rightarrow \mathbb{Z}/2\mathbb{Z}$, where $\beta : 1_G \mapsto 2\mathbb{Z}$ and $\beta : (gh) \mapsto 1 + 2\mathbb{Z}$. It is easy to check that β is a well-defined homomorphism, and it follows that $H \cong \mathbb{Z}/2\mathbb{Z}$. Note that despite being isomorphic, the operations in H and $\mathbb{Z}/2\mathbb{Z}$ are not the same; in H we multiply, while in $\mathbb{Z}/2\mathbb{Z}$ we add. However, the properties of homomorphisms ensure that these operations behave equivalently in each respective group.

Example 2.3.3. In example 2.2.6 we found that

$$G/H = \{H, (ab)H, (bc)H, (ac)H, (abc)H, (acb)H\},$$

and stated that it was a group. We can now formalize this by noting that $H \trianglelefteq G$. In fact, we can say more about G/H than just this; upon further inspection, the elements in G/H behave very similarly to the elements in S_3 . We can define a bijective mapping $\gamma : G/H \rightarrow S_3$ as follows:

$$\gamma : \begin{cases} H \mapsto (1) \\ (ab)H \mapsto (12) \\ (bc)H \mapsto (23) \\ (ac)H \mapsto (13) \\ (abc)H \mapsto (123) \\ (acb)H \mapsto (132) \end{cases}.$$

It is routine to check that γ is a well-defined homomorphism. Thus $G/H \cong S_3$.

We might wonder what occurs when we have a mapping from G to itself. In the case that it is an isomorphism, we call this an *automorphism*.

2.3.1 The Natural Map

Now that we know that the cosets of any normal subgroup, N , of G , form a group, G/N , and that homomorphisms are structure-preserving mappings between groups, we may want to define a homomorphism that moves between G and G/N . In fact, there is a special homomorphism that maps $G \rightarrow G/N$, which we call the *natural map*.

Definition 2.3.4. We define the *natural map* as the homomorphism from G to its quotient group:

$$\alpha : G \rightarrow G/N$$

where every $g \in G$ is mapped to its left coset

$$\alpha : g \mapsto gN.$$

Any two elements in the same coset of G/N are thus mapped to the same element by α .

2.3.2 The First Isomorphism Theorem

The natural map has profound implications for the way that we understand group homomorphisms. In fact, every group homomorphism is intimately linked to the natural map by Theorem 2.3.5.

Theorem 2.3.5. Let $\beta : G \rightarrow H$ be a group homomorphism. Then $\text{Im}(\beta) \leq H$, $\ker(\beta) \trianglelefteq G$, and $\text{Im}(\beta) \cong G/\ker(\beta)$.

We can understand Theorem 2.3.5 precisely by the natural map; using the natural map, we “mod out” the elements which are trivially mapped by β , leaving only those elements which appear in the image. Theorem 2.3.5 tells us that every normal subgroup is the kernel of a group homomorphism. Thus the natural map in this case is $\alpha : G \rightarrow G/\ker(\beta)$. In this way, we can then define a bijection $\hat{\beta} : G/\ker(\beta) \rightarrow H$, so that $\beta = \hat{\beta} \circ \alpha$. The mapping diagram is shown below.

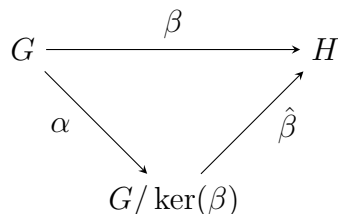


Figure 2.3.6: A group homomorphism, where $\hat{\beta}$ is a bijection.

Example 2.3.7. Returning once more to our poset in Figure 2.0.1, we can illustrate Theorem 2.3.5 by defining a homomorphism, β , from our group G to S_3 . Recall that we have defined an isomorphism, $\gamma : G/H \rightarrow S_3$. The natural map from G to G/H is $\alpha : G \rightarrow G/H$. Theorem 2.3.5 thus tells us that $\beta = \gamma \circ \alpha$, and that $\ker(\beta) = H$.

2.4 Group Actions

Thus far, we have studied the way groups behave on their own and when being mapped to other groups. The study of groups gains its full richness, however, when we consider the way a group acts upon a set. This is called a group action.

Definition 2.4.1. Let G be a group and X be a set. Then an *action* of G on X is a group homomorphism

$$\beta : G \rightarrow \text{Sym}(X)$$

where

$$\beta : g \mapsto \phi_g.$$

A group action of G on X can be stated equivalently as a mapping, λ , from $G \times X \rightarrow X$. We can define the *left action* of G on X ,

$$\lambda : G \times X \rightarrow X$$

where for $g \in G$, $x \in X$, $\lambda : g \mapsto g \cdot x$, such that the following properties hold:

1. The identity of G , 1_G acts trivially on every $x \in X$: $1_G \cdot x = x$.
2. The action is associative: for $g_1, g_2 \in G$ and $x \in X$, $(g_1 g_2) \cdot x = g_1 \cdot (g_2 \cdot x)$.

The *right action* is defined analogously and written $x \cdot g$.

Note that this second formulation is equivalent to Definition 2.4.1. For any $x \in X$ and $\phi_g \in \text{Sym}(X)$, $\phi_g(x) = g \cdot x$. Moreover, requiring that the identity act trivially and the action be associative is ensured in Definition 2.4.1 because β is a homomorphism.

2.4.1 Orbits and Stabilizers

Because a group action is a homomorphism between a specific pair of groups, the properties of group homomorphisms still apply. For $\beta : G \rightarrow \text{Sym}(X)$, $g \in \ker(\beta)$ if and only if $\beta : g \mapsto 1_{\text{Sym}(X)}$, the identity element of $\text{Sym}(X)$. However, we must now incorporate the fact that $1_{\text{Sym}(X)} = id_X$, the identity function on X . Thus

$$\ker(\beta) = \{g \in G \mid \beta(g) = 1_{\text{Sym}(X)}\}.$$

In addition to the kernel and image of the action, we may be interested in the effect the action has on any given element in the set being acted upon. Below, we define the *stabilizer* of $x \in X$, which is the set of group elements acting trivially upon x , and the *orbit* of x , which defines the possible elements x can be brought to within X by G .

Definition 2.4.2. For $x \in X$ we define the *stabilizer* of x to be the subgroup of G :

$$G_x = \{g \in G : g \cdot x = x\}.$$

The *orbit* of x is the set:

$$\text{Orb}(x) = \{y \in X : g \cdot x = y \text{ for some } g \in G\}.$$

Just as the set of cosets partition a group, the set of orbits partition the set upon which a group acts. However, the partitions formed by the orbit need not be the same size, as different set elements can react differently to the action at hand.

Example 2.4.3. Consider, one last time, our poset in Figure 2.0.1. The underlying set of this poset, X , is

$$X = \{a, b, c, d, e, f, g, h\}.$$

The permutations in G are acting upon X , and we can define a homomorphism, $\beta : G \rightarrow \text{Sym}(X)$. There are 5 distinct orbits of X ,

$$\text{Orb}(a) = \{a, b, c\},$$

$$\text{Orb}(g) = \{g, h\},$$

$$\text{Orb}(f) = \{f\},$$

$$\text{Orb}(e) = \{e\},$$

and

$$\text{Orb}(d) = \{d\},$$

which clearly partition X . In first discussing this example, we had somewhat loosely said that a was equivalent to b and c . The orbit gives us the vocabulary to formalize this, as we consider an orbit to be an equivalence relation between elements in a set.

There is a close link between orbits, which characterize the path of an element under the action of a group, and the stabilizer, which enumerates the group elements which “stabilize” a given set element.

Theorem 2.4.4 (Orbit-Stabilizer). *Let G be a group acting on a set X . Then for $x \in X$,*

$$|\text{Orb}(x)| = |G : G_x| = \frac{|G|}{|G_x|}.$$

This should look extremely similar to Theorem 2.2.4, where we had $|G| = |G : H| \cdot |H|$. In fact, one way to understand Theorem 2.4.4 is as a generalization of Theorem 2.2.4, in which a group G acts on the set G/H . In this case, the stabilizer of the action is H and any orbit is equal to G/H .

2.4.2 Determining the Size of X

We find the order of X by summing over its orbits, which partition X . If G acts on a finite set X with t distinct orbits x_i for $1 \leq i \leq t$, the order of X is

$$|X| = \sum_{i=1}^t |G : G_{x_i}|. \quad (2.4.5)$$

If every element in a set can be sent to any other element by the action of some element in a group, then the action has a single orbit. We say that this is a *transitive* group action.

Definition 2.4.6. The action of a group G on a set X is *transitive* if for every $x, y \in X$, there exists $g \in G$ such that $g \cdot x = y$.

Determining the number of distinct orbits formed under the action is closely linked to determining the elements in X for which some $g \in G$ acts trivially. These set elements are the *fixed points* of g .

Definition 2.4.7. For a finite group G acting on X , the *fixed points* of $g \in G$ are

$$\text{Fix}(g) = \{x \in X | g \cdot x = x\}.$$

If every non-trivial element in a group has no fixed points in a group action, we say that the action is *free*.

Definition 2.4.8. An action G on a set X is *free* if for every $x \in X$, if $g \cdot x = x$ then $g = 1_G$. An action is *regular* if it is free and transitive.

We can use the properties of transitivity and freeness to determine the size of X with a standard theorem from the study of groups.

Theorem 2.4.9. For a finite set X , the action of G on X is transitive if and only if for every $x \in X$, $|G : G_x| = |X|$. The action of G on X is regular if and only if the action is transitive and $|X| = |G|$.

This follows from the fact that the action of G on X has a single orbit if and only if it is transitive. A regular action on a finite set has the special property that $G_x = 1_G$.

Of course, our primary interest in this thesis is on posets. With the knowledge we have built in this chapter, we may now proceed to the study of the symmetries of posets and the way they relate to linear extensions.

Chapter 3

Poset Symmetry

Having thus provided an introduction to the study of groups, we return once more to posets. Using the material covered in Chapters 1 and 2, we can delve into the study of the automorphism group of a poset and its relationship to its linear extensions. In addition, we discuss the dual of a poset and the quotient poset, and provide several original proofs that relate these entities.

3.1 The Automorphism Group of a Poset

3.1.1 Order Embeddings

The permutations in $\text{Sym}(X)$ permute the elements in X without reference to any order relations in P . Therefore, if we want to ensure that any permutation we apply to P preserves the order in a poset, we must be more picky about the permutations we deem acceptable. For this reason, we introduce the following order-conscious mappings.

Definition 3.1.1. Let $P_1 = (X, \leq)$ and $P_2 = (Y, \leq)$ be posets, and $\gamma : X \rightarrow Y$ a function. We say that γ is an *order preserving map* if for every $x, y \in X$, if $x < y$ in P_1 , then $\gamma(x) < \gamma(y)$ in P_2 . We say that γ is an *order reflecting map* if for every $x, y \in X$, if $\gamma(x) < \gamma(y)$ in P_2 , then $x < y$ in P_1 . If γ is order preserving and order reflecting, then γ is an *order embedding* of P_1 into P_2 .

Order embeddings need not be bijective. However, we are particularly interested in the maps that are.

Definition 3.1.2. Two posets P_1 and P_2 are isomorphic if there exists some bijective order embedding map $\gamma : P_1 \rightarrow P_2$. In this case, we write $P_1 \cong P_2$, and define γ to be a *poset isomorphism*. If $\gamma : P \rightarrow P$ is a bijective order embedding, then we say that γ is a *poset automorphism*.

The collection of these mappings forms a group, which we denote as $\text{aut}(P)$, the *automorphism group of P* .

The automorphism group of P is made up of the permutations in X that preserve the order relations in P , and they form a subgroup of $\text{Sym}(X)$. We have secretly

encountered the automorphism group before: the group of permutations, G , that we studied extensively in Chapter 2 from the poset in Figure 2.0.1 was actually the automorphism group of that poset.

One way of understanding the relationship between $\text{aut}(P)$ and $\text{Sym}(X)$ in reference to a sorting sequence, P_0, P_1, \dots, P_n , is that initially, we begin with an unordered set and so $\text{aut}(P_0) = \text{Sym}(X)$. As we begin to make comparisons, order relations are established and so $\text{aut}(P_j) < \text{Sym}(X)$ for $1 \leq j \leq n$. That is, $\text{aut}(P_j)$ changes as we go from P_0 to P_n , but $\text{Sym}(X)$ never does because we work with the same underlying set throughout a sorting sequence. We end with a total ordering, where $\text{aut}(P_n) = \{1_{\text{Sym}(X)}\}$.

3.1.2 Order Reversing Maps

Just as we have defined maps that preserve the order of a poset, we can define a class of mappings that reverses the order of a poset.

Definition 3.1.3. Let $P_1 = (X, \leq)$ and $P_2 = (Y, \leq)$ be posets, and $\gamma^* : P_1 \rightarrow P_2$. We say that γ^* is an *order reversing map* if for every $x, y \in X$, if $x < y$ in P_1 , then $\gamma(x) > \gamma(y)$ in P_2 . We say that γ^* is a *reverse order reflecting map* if for every $x, y \in X$, if $\gamma(x) < \gamma(y)$ in P_2 , then $x > y$ in P_1 . If γ^* is order reversing and reverse order reflecting, then γ^* is a *reverse order embedding* of P_1 into P_2 .

Like order embeddings, reverse order embeddings need not be bijective—but once again, we are particularly interested in those that are. Two posets P_1 and P_2 are *anti-isomorphic* if there exists a bijective reverse order embedding map $\gamma^* : P_1 \rightarrow P_2$. Naturally, an *anti-automorphism* of a poset, $P = (X, \leq)$, is a bijective reverse order embedding that maps $X \rightarrow X$, but reverses every ordering in P . We shall denote this set of reverse orders as \leq^* . Thus an anti-automorphism of P is a bijective order embedding $(X, \leq) \rightarrow (X, \leq^*)$. We refer to the poset $P^* = (X, \leq^*)$ as the *dual* of P .

Definition 3.1.4. For a poset $P = (X, \leq)$, the *dual* of P is the poset

$$P^* = (X, \leq^*)$$

such that for every $x, y \in X$, $x < y$ in P if and only if $x > y$ in P^* .

The dual provides an alternative definition of anti-isomorphism; P_1 and P_2 are anti-isomorphic if there exists a bijective order embedding $\gamma : P_1 \rightarrow P_2^*$. When $P \cong P^*$, we say that P is *self-dual*. In this case, an anti-automorphism of P is also an isomorphism between P and P^* . A self-dual poset P has a *midpoint*, x , if there exists some anti-automorphism of P that fixes x . In order to distinguish this anti-automorphism from others, if P is a self-dual poset with midpoints, we will denote $\delta_x : P \rightarrow P^*$ to be any anti-isomorphism that fixes the midpoint x .

3.1.3 The Quotient Poset

Orbits form an equivalence relation, and we can gain a deeper understanding of our poset P and $\text{aut}(P)$ by grouping its equivalence classes together. Just as we can form

a group by modding out a normal subgroup via the natural map, we can form a new poset from by modding out the orbits of P . We call this poset the *quotient poset of P* , and will see shortly that it is a powerful tool in understanding how $\text{aut}(P)$ interacts with $\varepsilon(P)$.

Defining the Quotient Poset

Let $P = (X, \leq)$ be a poset. In order to define the quotient poset, we must first define its underlying set. We call this set X/\sim . For every $x \in X$, we define $[x] \in X/\sim$ to be $[x] = \text{Orb}(x)$. Thus for $x, z \in P$, $[x] = [z]$ in X/\sim if there is some $\pi \in \text{aut}(P)$ such that $x = \pi \cdot z$. This is equivalent to z and x being in the same orbit.

Because we wish to define the quotient *poset*, we must introduce a notion of order in X/\sim .

Definition 3.1.5. Let $P = (X, \leq)$. We say that $[x] \leq [y]$ in X/\sim if there exists a $\pi \in \text{aut}(P)$ such that $x \leq \pi \cdot y$ in P .

We can now show that this notion of order on the set X/\sim produces a poset, which we define to be the *quotient poset*.

Proposition 3.1.6. Let $P = (X, \leq)$ be a poset. Then $P/\sim = (X/\sim, \leq)$ is a poset.

Proof. Let $x, y, z, w \in P$. We check the following conditions:

1. P/\sim is well-defined. Let $[x] = [w]$ and $[y] = [z]$. Suppose that $[x] \leq [y]$. Then there is $\sigma \in \text{aut}(P)$ such that $x \leq \sigma \cdot y$ and $\tau, \pi \in \text{aut}(P)$ such that $\tau \cdot w = x$ and $\pi \cdot z = y$. Thus $\tau \cdot w \leq \sigma \cdot (\pi \cdot z)$. So $w \leq (\tau^{-1}(\sigma\pi)) \cdot z$, and therefore $[w] \leq [z]$. The other direction is symmetric.
2. P/\sim is reflexive. Clearly $x \leq x$. So $x \leq 1_{\text{aut}(P)} \cdot x$. Thus $[x] \leq [x]$.
3. P/\sim is transitive. Suppose $[x] \leq [y]$ and $[y] \leq [z]$. So there are $\sigma, \tau \in \text{aut}(P)$ such that $x \leq \sigma \cdot y$ and $y \leq \tau \cdot z$. Therefore $\sigma \cdot y \leq \sigma \cdot (\tau \cdot z)$, and so $x \leq (\sigma\tau) \cdot z$. Thus $[x] \leq [z]$.
4. P/\sim is anti-symmetric. Suppose $[x] \leq [y]$ and $[y] \leq [x]$. Then there are $\sigma, \tau \in \text{aut}(P)$ such that $x \leq \sigma \cdot y$ and $y \leq \tau \cdot x$. So $\tau^{-1} \cdot y \leq x \leq \sigma \cdot y$. Note that two elements in the same orbit are either equal or incomparable. Since $\tau^{-1} \cdot y \leq \sigma \cdot y$, we know that $\tau^{-1} \cdot y = \sigma \cdot y$. So $x = \tau^{-1} \cdot y = \sigma \cdot y$. Hence $[x] = [y]$.

□

Thus P/\sim is a poset whose size is equal to the number of orbits in P . Interestingly, even though we mod out $\text{aut}(P)$, the automorphism group of P/\sim need not be trivial.

Example 3.1.7. Recall the poset first seen in Figure 2.0.1, which we shall call P . We determine its quotient poset, as shown in Figure 3.1.8. As we can see, $\text{aut}(P/\sim)$ is not trivial.

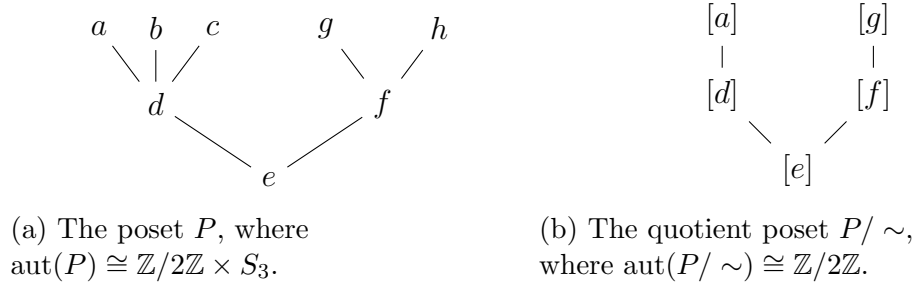


Figure 3.1.8: A Poset and its Quotient Poset.

The Natural Poset Map and Its Pull-Back

Having thus defined the quotient poset, we can determine a map that moves from P to P/\sim .

Definition 3.1.9. The *natural poset map* Ω , is a mapping

$$\Omega : X \rightarrow X/\sim$$

where

$$\Omega : x \mapsto [x].$$

It should be clear that Ω is a surjective, order-preserving map. For any x, y in P , if $x \leq y$ in P , then $x \leq 1_{\text{aut}(P)} \cdot y$, so $[x] \leq [y]$ in P/\sim . Moreover, for $[x] \in P/\sim$, $x = 1_{\text{aut}(P)} \cdot x$, so $[x] \in \text{Im}(\Omega)$.

The fact that Ω is order preserving and surjective means that there is an important relationship between $\varepsilon(P/\sim)$ and $\varepsilon(P)$. In particular, every linear extension of P/\sim can be turned into a subset of the linear extensions of P by looking at the pull-back of $M \in \varepsilon(P/\sim)$ by Ω . For $M \in \varepsilon(P/\sim)$,

$$M = ([x_1], [x_2], \dots, [x_k]),$$

we can determine the the subset of linear extensions of P obtained by the reverse image of Ω as follows. Because $|M| = k$, there are k distinct orbits in P . We can pull-back each $[x_i]$,

$$\Omega^{-1}([x_i]) = \text{Orb}_i \text{ where } 1 \leq i \leq k.$$

Let $\varepsilon(\text{Orb}_i)$ denote the set of linear extensions for the poset Orb_i , which is discrete because all of its elements lie in the same orbit. We thus define the pull-back of M by Ω ,

$$\Omega^{-1}(M) = \{L : L = L_1 \boxplus L_2 \boxplus \dots \boxplus L_k \text{ where each } L_i \in \varepsilon(\text{Orb}_i)\}.$$

Note that $\Omega^{-1}(M)$ is the set of linear extensions of a serial compositions of orbits, the order of which is determined by M . Each of the orbits is a discrete poset, and it thus follows that

$$\Omega^{-1}(M) = \varepsilon(\text{Orb}_1 \boxplus \text{Orb}_2 \boxplus \dots \boxplus \text{Orb}_k). \quad (3.1.10)$$

The poset $Q = \text{Orb}_1 \boxplus \text{Orb}_2 \boxplus \dots \boxplus \text{Orb}_k$ has the same underlying set as P and if $x \leq y$ in P , then $x \leq y$ in Q . Thus P is a subset of Q , where $|P| = |Q|$. It follows that $\varepsilon(Q) \subseteq \varepsilon(P)$. We call this subset of linear extensions of P , which is determined by M , $\varepsilon_M(P)$. We can precisely determine the size of $\varepsilon_M(P)$ by Equation 3.1.10,

$$e_M(P) = |\varepsilon_M(P)| = \prod_{i=1}^k |\text{Orb}_i|!. \quad (3.1.11)$$

We can similarly pull back the set of linear extensions of $\varepsilon(P/\sim)$, which we call $\varepsilon_\Omega(P)$:

$$\varepsilon_\Omega(P) = \bigcup_{M \in \varepsilon(P/\sim)} \varepsilon_M(P).$$

By the same logic as before, $\varepsilon_\Omega(P) \subseteq \varepsilon(P)$. Every $M \in \varepsilon(P/\sim)$ rearranges the orbits differently, but importantly, $e_M(P)$ is always the same for any M . The size of $\varepsilon_\Omega(P)$ is thus

$$e_\Omega(P) = e(P/\sim) \cdot \prod_{i=1}^k |\text{Orb}_i|!. \quad (3.1.12)$$

Equation 3.1.12 therefore gives a lower bound on the number of linear extensions for any poset:

$$e(P) \geq e(P/\sim) \cdot \prod_{i=1}^k |\text{Orb}_i|!.$$

3.2 Linear Extensions and the Automorphism Group

The automorphism of a poset naturally acts on the poset itself. In this section, we define a new group action: that of $\text{aut}(P)$ on $\varepsilon(P)$.

3.2.1 The Action of $\text{aut}(P)$ on $\varepsilon(P)$

We can understand something about the relationship between the automorphism group of a poset, $\text{aut}(P)$, and its linear extensions, $\varepsilon(P)$, by examining how $\text{aut}(P)$ acts on $\varepsilon(P)$. We define the action of $\text{aut}(P)$ on $\varepsilon(P)$ by the way it permutes each element of the ordering in a linear extension.

Definition 3.2.1. Let $|P| = n$. For any $L \in \varepsilon(P)$, $L = (x_1, x_2, \dots, x_n)$, and any $\pi \in \text{aut}(P)$, π acts on L by

$$\pi \cdot L = (\pi \cdot x_1, \pi \cdot x_2, \dots, \pi \cdot x_n).$$

It is straightforward to show that this action is well-defined because by definition, $\text{aut}(P)$ is the collection of order preserving permutations and for distinct elements $x, y \in P$, if $\pi \cdot x = y$, then x and y must be incomparable. Thus $\pi \cdot L \in \varepsilon(P)$.

The action of $\text{aut}(P)$ on $\varepsilon(P)$ allows us to relate the size of any orbit in P , as well as the size of $\text{aut}(P)$, to $e(P)$.

Theorem 3.2.2. For a poset P with $x \in P$, $|\text{Orb}(x)|$ divides $e(P)$. Moreover, $|\text{aut}(P)|$ divides $e(P)$.

The proof of Theorem 3.2.2 relies heavily on the following Lemma, which we prove first. Recall that a group action is said to be free if every non-trivial element in the group has no fixed points.

Lemma 3.2.3. For a poset P , the action of $\text{aut}(P)$ on $\varepsilon(P)$ is free.

Proof. Let L be an arbitrary linear extension of P . Suppose that $\pi \in \text{aut}(P)$ is not the identity element for $\text{aut}(P)$. Then π is the product of disjoint cycles; choose any one of these: $(x_1x_2\dots x_n)$ for $n \geq 2$. Consider the configuration of x_1, x_2, \dots, x_n in L , and label the cycle element that appears lowest in L as x_1 . Then $x_1 < x_2$ and $x_1 < x_n$ in L (if $n = 2$, then $x_2 = x_n$). Applying π to L , we have $\pi \cdot x_1 = x_2$ and $\pi \cdot x_n = x_1$, so $x_2 < x_1$ in $\pi \cdot L$. Thus $\pi \cdot L \neq L$. \square

Lemma 3.2.3 thus tells us that any non-trivial $\pi \in \text{aut}(P)$ brings $L \in \varepsilon(P)$ to a different linear extension. We can now prove Theorem 3.2.2.

Proof of Theorem 3.2.2. Consider the action of $\text{aut}(P)$ on $\varepsilon(P)$. By Lemma 3.2.3, this action is free, and therefore has no non-trivial stabilizers. By the Orbit-Stabilizer Theorem, $|\text{aut}(P)| = |\text{Orb}(L)|$ for any $L \in \varepsilon(P)$. Because this is true of an arbitrary L , we know that the action of $\text{aut}(P)$ on $\varepsilon(P)$ partitions $\varepsilon(P)$ into subsets of size $|\text{Orb}(L)|$, and so $|\text{Orb}(L)|$ divides $e(P)$. Therefore, $|\text{aut}(P)|$ divides $e(P)$. Now take $x \in P$. Applying the Orbit-Stabilizer Theorem again, we have that $|\text{Orb}(x)|$ divides $|\text{aut}(P)|$. So $|\text{Orb}(x)|$ divides $e(P)$. \square

3.2.2 When $e(P) = |\text{aut}(P)|$

The order of $\text{aut}(P)$ divides $e(P)$ for any poset; we may wonder if there are classes of posets for which $|\text{aut}(P)| = e(P)$. Recall that by Theorem 2.4.9, for a group G acting upon a finite set, X , the action of G on X is regular (free and transitive) if and only if the action is transitive and $|G| = |X|$. By Lemma 3.2.3, the action of $\text{aut}(P)$ on $\varepsilon(P)$ is always free; thus $e(P) = |\text{aut}(P)|$ if and only if the action of $\text{aut}(P)$ on $\varepsilon(P)$ is transitive.

Transitivity ensures that any element in the set can be sent to any other element by some element in the group. Thus for a poset P , $\text{aut}(P)$ acting on $\varepsilon(P)$ is transitive if for any $L_1, L_2 \in \varepsilon(P)$, there is a unique $\pi \in \text{aut}(P)$ that sends L_1 to L_2 . This property is clearly not true for all posets, and a natural question, therefore, is for which classes of posets it does hold.

We first define a class of posets, and then prove that this is the exactly class of posets for which $|\text{aut}(P)| = e(P)$.

Definition 3.2.4. A poset P with height h is *pointwise decomposable* if it can be serially decomposed into h subposets, each of which is a discrete poset.

An important property of pointwise decomposable posets is that every pair of incomparable elements must be in each other's orbits. We denote the serial decompositions of a pointwise decomposable poset P as ω_i for $1 \leq i \leq h$, where for $x \in \omega_i$ and $y \in \omega_{i+1}$, $x < y$. We denote the number of elements in ω_i as $|\omega_i|$.

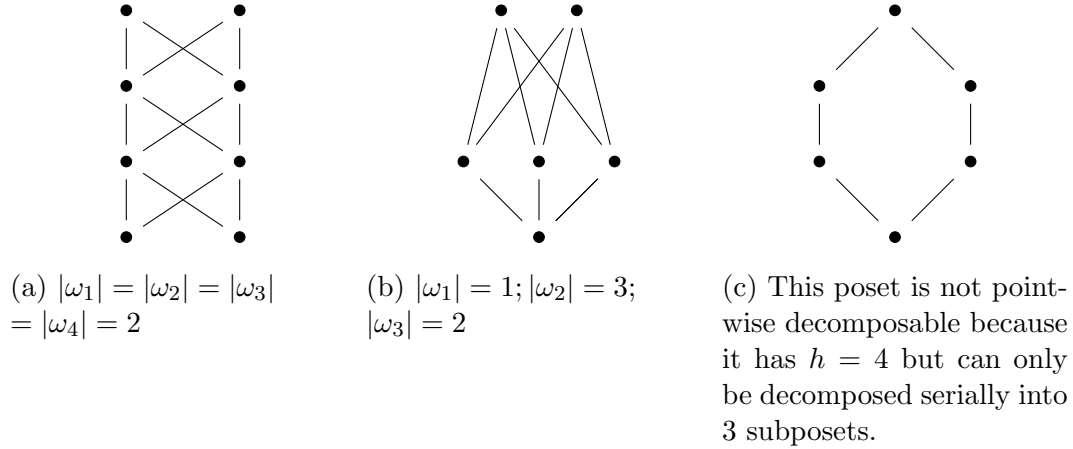


Figure 3.2.5: The first two posets are pointwise decomposable; the last is not.

We now prove that this is the only class of posets in which $\text{aut}(P) = e(P)$ by showing that pointwise decomposable posets are the only class of posets where $\text{aut}(P)$ acts transitively on $e(P)$.

Theorem 3.2.6. *A poset P is pointwise decomposable if and only if the action of $\text{aut}(P)$ on $\varepsilon(P)$ is transitive.*

Proof. Suppose P is pointwise decomposable. Then it can be serially decomposed into h subposets $\omega_1 \boxplus \dots \boxplus \omega_h$. Because each ω_i is discrete, there are $|\omega_i|!$ ways to sort each partition. Thus

$$e(P) = \prod_{i=1}^h |\omega_i|!$$

Consider $\text{aut}(P)$. Every ω_i is made up of discrete, serially composed posets, so

$$\text{aut}(P) = \text{Sym}(\omega_1) \times \text{Sym}(\omega_2) \times \dots \times \text{Sym}(\omega_h).$$

We thus get

$$|\text{aut}(P)| = \prod_{i=1}^h |\text{Sym}(\omega_i)| = \prod_{i=1}^h |\omega_i|!$$

So $e(P) = |\text{aut}(P)|$, and therefore P is transitive.

Conversely, suppose $\text{aut}(P)$ acts transitively on $\varepsilon(P)$. We form the quotient poset of P , P/\sim , and take any $M \in \varepsilon(P/\sim)$. We can then obtain the subset of $\varepsilon(P)$, $\varepsilon_M(P)$, by pulling back M by the natural poset map Ω .

We show that the action of $\text{aut}(P)$ on $\varepsilon(P)$ restricts to $\varepsilon_M(P)$.

To this end, take $\pi \in \text{aut}(P)$ and $L \in \varepsilon_M(P)$, and apply π to L . Clearly, π only permutes elements that are in the same orbit, and by assumption L is of the form $L_1 \boxplus L_2 \boxplus \dots \boxplus L_k$, where $L_i \in \varepsilon(\text{Orb}_i)$. Thus

$$\pi \cdot L = \pi \cdot L_1 \boxplus \pi \cdot L_2 \boxplus \dots \boxplus \pi \cdot L_k,$$

which is in $\varepsilon_M(P)$. So the action of $\text{aut}(P)$ on $\varepsilon(P)$ restricts to $\varepsilon_M(P)$. Therefore the action of $\text{aut}(P)$ on $\varepsilon_M(P)$ inherits both the freeness and transitivity of the action of $\text{aut}(P)$ on $\varepsilon(P)$. So $e_M(P) = |\text{aut}(P)|$, and thus $e_M(P) = e(P)$.

Because $\varepsilon_M(P) \subseteq \varepsilon(P)$, it follows that

$$\varepsilon_M(P) = \varepsilon(P).$$

By Equation 3.1.10,

$$\varepsilon(P) = \varepsilon_M(P) = \varepsilon(\text{Orb}_1 \boxplus \text{Orb}_2 \boxplus \dots \boxplus \text{Orb}_k),$$

and so because P is a subposet of $\text{Orb}_1 \boxplus \text{Orb}_2 \boxplus \dots \boxplus \text{Orb}_k$ with the same set of linear extensions, we conclude that

$$P = \text{Orb}_1 \boxplus \text{Orb}_2 \boxplus \dots \boxplus \text{Orb}_k.$$

Thus P must be pointwise decomposable. \square

We can now update our theorem about the case when $e(P) = |\text{aut}(P)|$.

Corollary 3.2.7. *Let P be a finite poset. Then P is pointwise decomposable if and only if $e(P) = |\text{aut}(P)|$.*

One way of understanding the pointwise decomposable poset is as the only class of poset whose set of linear extensions act harmoniously with the poset's automorphisms. If anything, the specificity of the pointwise decomposable posets affirms what we already knew: that linear extensions are generally very complicated to determine!

3.2.3 When $e(P)$ is prime

Finally, we can combine these theorems to show that if $e(P)$ is an odd prime, we can completely determine its group of symmetries (or lack thereof).

Theorem 3.2.8. *If P is a poset with $e(P) = p$, where p is prime and $p \neq 2$, then $\text{aut}(P)$ is trivial.*

Proof. Let $e(P) = p$. Then by Theorem 3.2.2, $|\text{aut}(P)|$ divides $e(P)$, so $|\text{aut}(P)| = p$ or $|\text{aut}(P)| = 1$. Suppose $|\text{aut}(P)| = p = e(P)$. Then by Theorem 3.2.6, P is pointwise decomposable. Thus

$$e(P) = \prod_{i=1}^h |\omega_i|!.$$

Note that the only prime number this formula can yield is 2. But $p \neq 2$. So $|\text{aut}(P)| = 1$, and therefore must be the trivial group. \square

We might wonder whether any posets actually have a prime number of linear extensions. It turns out we need look no further than our favorite poset from Chapter 1, the N poset!

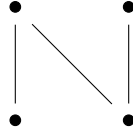


Figure 3.2.9: An N poset, which has 5 linear extensions.

3.3 Perfect Splits

The information-theoretic lower bound is based upon the fact that the best case comparison in P perfectly divides $\varepsilon(P)$ in half. In this section, we characterize some of the structural features of a poset that allow such a split to occur.

First, we prove a lemma which will be useful in our proofs. Recall that we say $P \sim Q$ up to serial parallel composition if Q is a subposet of P that can be obtained from a series of \boxplus and \oplus decompositions to P .

Lemma 3.3.1. *Let P be a poset with a subposet Q where $P \sim Q$ up to serial parallel composition. Then for incomparable elements $x, y \in Q$, $\chi(Q, x, y) = \chi(P, x, y)$.*

Proof. Let $x, y \in Q$ be incomparable. Without loss of generality let $e(Q_{x < y}) \geq e(Q_{y < x})$, so

$$\chi(Q, x, y) = \frac{e(Q_{x < y})}{e(Q)}.$$

We proceed by induction. In the base case there are two possibilities.

1. Suppose that $P = Q \boxplus S$. Then $e(P) = e(Q)e(S)$. Now compare some x, y in Q . We get that $P_{x < y} = Q_{x < y} \boxplus S$, and

$$e(P_{x < y}) = e(Q_{x < y})e(S) = \chi(Q, x, y)e(Q)e(S) = \chi(Q, x, y)e(P).$$

Thus

$$\chi(P, x, y) = \frac{e(P_{x < y})}{e(P)} = \chi(Q, x, y).$$

2. Now suppose that $P = Q \oplus S$, so $P_{x < y} = Q_{x < y} \oplus S$, and

$$e(P) = e(Q)e(S) \binom{|Q| + |S|}{|S|}.$$

Again, compare $x, y \in Q$. Note that $|Q| = |Q_{x < y}|$, so

$$e(P_{x < y}) = e(Q_{x < y})e(S) \binom{|Q| + |S|}{|S|} = \chi(Q, x, y)e(P),$$

and therefore $\chi(Q, x, y) = \chi(P, x, y)$.

The inductive step follows immediately from the fact that \boxplus and \oplus are binary operations. \square

3.3.1 When $|\text{aut}(P)|$ is even

We have used the fact that distinct elements in the same orbit in P are necessarily incomparable. Recall that in the case that there is a permutation π in $\text{aut}(P)$ that swaps a pair of elements x and y in the same orbit, we say that π *transposes* x and y . We now prove that having such a π is a sufficient condition to guarantee that $P_{x < y} \cong P_{y < x}$ and thus that $\chi(P, x, y) = \frac{1}{2}$.

Theorem 3.3.2. *Let P be a poset with $\pi \in \text{aut}(P)$ such that π transposes two elements x and y in P . Then $\chi(P, x, y) = \frac{1}{2}$.*

Proof. Let $\pi \in \text{aut}(P)$ transpose x and y . We define the set $\Sigma_{x,y}$ as

$$\Sigma_{x,y} = \{\omega : \omega \in P, \omega \perp x, \text{ and } \omega || y\}.$$

Note that because π transposes x and y , we have that $\pi : \Sigma_{x,y} \rightarrow \Sigma_{y,x}$ is a bijection, and similarly, $\pi : \Sigma_{y,x} \rightarrow \Sigma_{x,y}$ is a bijection. For any $z \in P$ where $z \notin \Sigma_{x,y} \cup \Sigma_{y,x}$, it follows that $\pi \cdot z \notin \Sigma_{x,y} \cup \Sigma_{y,x}$.

Consider $P_{x < y}$. The posets P and $P_{x < y}$ differ only in the relationship between x and y and any order relations determined by transitivity upon comparing x and y , which must be of the form

$$\{\sigma < y : \sigma \in \Sigma_{x,y} \text{ and } \sigma < x \text{ in } P\}$$

and

$$\{\sigma > x : \sigma \in \Sigma_{y,x} \text{ and } \sigma > y \text{ in } P\}.$$

For $P_{y < x}$ this is exactly reversed. By applying π to $P_{x < y}$, we see that it maps to $P_{y < x}$. Thus π is a bijective order embedding from $P_{x < y} \rightarrow P_{y < x}$.

So $P_{x < y} \cong P_{y < x}$ and therefore $e(P_{x < y}) = e(P_{y < x})$. \square

Recall by Cauchy's Theorem (Theorem 2.1.5), for any finite group G , if a prime number p divides $|G|$, then there is an element of order p in G . If $|\text{aut}(P)|$ is even, there must be an element π of order 2 in $\text{aut}(P)$. Because $|\pi| = 2$, it must have at least one non-trivial cycle of length 2; let x and y be the elements of that cycle. Then π transposes x and y , giving us the following corollary of Theorem 3.3.2.

Corollary 3.3.3. *Let P be a poset with an automorphism group $\text{aut}(P)$ of even order. Then there exist a pair of incomparable elements $x, y \in P$ such that $\chi(P, x, y) = \frac{1}{2}$.*

Theorem 3.3.2 is similar (though not identical) to one proved by Ganter, Häfner and Poguntke [7], but was discovered independently of it.

Our proof of Theorem 3.3.2 relied on having a permutation that transposed two elements. This property is not true of posets whose automorphism group is odd, and Theorem 3.3.2 does not hold for such posets.

Example 3.3.4. Consider the poset diagram in Figure 3.3.5, where $\text{aut}(P) \cong \mathbb{Z}/3\mathbb{Z}$ and $|\text{aut}(P)| = 3$. While $x \in \text{Orb}(y)$, when we compare x with y , we get two posets which are not isomorphic, as Figure 3.3.6 shows. To see why $P_{x < y} \not\cong P_{y < x}$, all we need note is that the maximal elements in $P_{x < y}$ have respective down-covers of size 2 and 3, while in $P_{y < x}$ both maximal elements have down-covers of size 2.

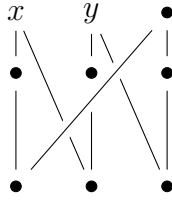


Figure 3.3.5: A Poset P where $\text{aut}(P) \cong \mathbb{Z}/3\mathbb{Z}$.

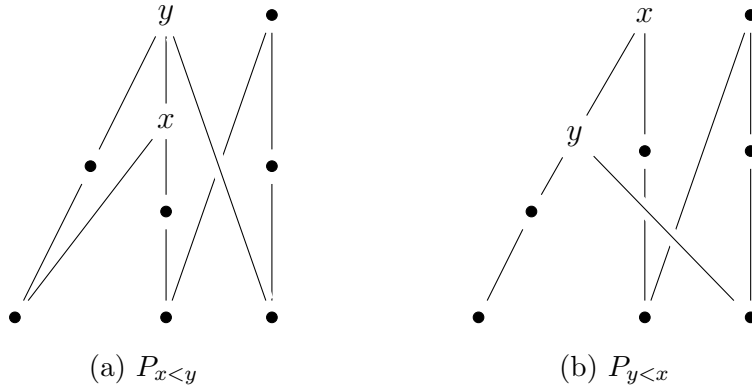


Figure 3.3.6: Comparing x with y does not yield isomorphic posets.

3.3.2 Self-Dual Posets

Having an automorphism of even order is a sufficient condition for a perfect split, but it is not necessary, as the next example shows.

Example 3.3.7. The first poset pictured in Figure 3.3.8 has 4 linear extensions; when we compare x with y , the resulting posets both have 2 linear extensions. They are not isomorphic—however, they *are* anti-isomorphic, and $\text{aut}(P_{x < y}) \cong \text{aut}(P_{y < x}) \cong \mathbb{Z}/2\mathbb{Z}$.

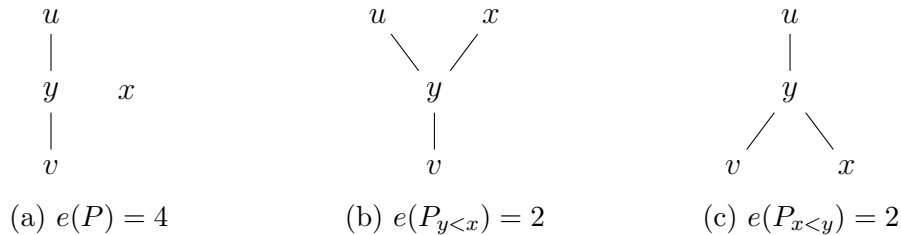


Figure 3.3.8: An anti-isomorphic perfect split.

Example 3.3.7 is an instance of another condition sufficient for a perfect split. Note that P is a parallel composition of two self-dual posets, with midpoints y and x respectively. We now prove that allows for a perfect split, and use Lemma 3.3.1 to make the condition more general.

Theorem 3.3.9. *Let P be a poset with a subposet Q such that $Q = P_1 \oplus P_2$ where $Q \sim P$ up to serial-parallel composition. If P_1 and P_2 are self-dual posets with midpoints x and y respectively, then $\chi(P, x, y) = \frac{1}{2}$.*

Proof. Let Q be as defined above. We show that $Q_{x<y} \cong Q_{y<x}^*$.

Because P_1 and P_2 are self-dual posets with midpoints, there exist well-defined, isomorphisms, $\delta_x : P_1 \rightarrow P_1^*$ and $\delta_y : P_2 \rightarrow P_2^*$, where δ_x fixes x and δ_y fixes y . We can thus define a mapping ν as follows:

$$\nu(z) = \begin{cases} \delta_x(z) & z \in P_1 \\ \delta_y(z) & z \in P_2 \end{cases}.$$

We show that ν is a poset isomorphism from $Q_{x<y}$ to $Q_{y<x}^*$.

Clearly $\nu : Q \rightarrow Q^*$ is bijective and order preserving. Thus we need only ensure that ν is order preserving when it maps the new order relations in $Q_{x<y}$ to $Q_{y<x}^*$. In particular, in $Q_{x<y}$ we learn that $x < y$ and by transitivity that

$$\{z < y : z \in D_Q(x)\}$$

and

$$\{z > x : z \in U_Q(y)\}.$$

In $Q_{y<x}^*$, $x < y$. By assumption x and y are midpoints and $\nu(x) = x$, $\nu(y) = y$, so $\nu(x) < \nu(y)$ in $Q_{y<x}^*$.

Let $z \in D_Q(x)$, so $z < x < y$ in $Q_{x<y}$. Because $\nu(D_Q(x)) = U_Q(x)$ it follows that $\nu(z) \in U_Q(x)$. In $Q_{y<x}^*$, for every $q \in U_Q(x)$, $q < x$. So $\nu(z) < \nu(x) < \nu(y)$ in $Q_{y<x}^*$.

For $z \in U_Q(y)$ the argument is symmetric, as is the argument for the reverse direction, where $\nu^{-1} : Q_{y<x}^* \rightarrow Q_{x<y}$. So $\nu : Q_{x<y} \rightarrow Q_{y<x}^*$ is a bijective order embedding. The theorem follows immediately, as $e(Q_{x<y}) = e(Q_{y<x}^*)$. □

3.3.3 Serial Parallel Cases

In Theorems 3.3.2 and 3.3.9, we were able to use the fact that an automorphism or anti-automorphism of a poset P could induce an isomorphism or anti-isomorphism between $P_{x<y}$ and $P_{y<x}$. We now turn to the cases where we can determine a perfect split by knowing how P , $P_{x<y}$ and $P_{y<x}$ are serially decomposed.

Theorem 3.3.10. *Let P be a poset with a subposet Q such that $Q = P_1 \oplus P_2$, with $Q \sim P$ up to serial-parallel composition. If $|P_1| = |P_2|$ and P_1 and P_2 have respective minima x and y , then $\chi(P, x, y) = \frac{1}{2}$. Dually, if x and y are the respective maxima of P_1 and P_2 , then $\chi(P, x, y) = \frac{1}{2}$.*

Proof. By Lemma 3.3.1, it is sufficient to show that $e(Q_{x<y}) = e(Q_{y<x})$. Let x and y be minima of P_1, P_2 , respectively. Then $e(P_1) = e(U_Q(x))$ and $e(P_2) = e(U_Q(y))$. Moreover, $\min(Q_{x<y}) = \{x\}$ and $\min(Q_{y<x}) = \{y\}$. Thus

$$Q_{x<y} = \{x\} \boxplus (U_Q(x) \oplus P_2)$$

and

$$Q_{y<x} = \{y\} \boxplus (U_Q(y) \oplus P_1),$$

so $e(Q_{x<y}) = e(U_Q(x) \oplus P_2)$ and $e(Q_{y<x}) = e(U_Q(y) \oplus P_1)$. It follows that

$$e(Q_{x<y}) = e(U_Q(x)) \cdot e(P_2) \cdot \binom{|U_Q(x)| + |P_2|}{|P_2|} = e(P_1) \cdot e(P_2) \cdot \binom{(|P_1| - 1) + |P_2|}{|P_2|},$$

and similarly,

$$e(Q_{y<x}) = e(P_1) \cdot e(P_2) \cdot \binom{|P_1| + (|P_2| - 1)}{|P_1|}.$$

Because $|P_1| = |P_2|$, these are equal. The case for x and y as maxima is symmetric, so we are done. \square

The next theorem determines a sufficient condition to have a perfect split in a poset that is not serial parallel but has a pair of elements whose comparison results in a serial composition.

Theorem 3.3.11. *Let P be a poset with a subposet Q , where $Q \sim P$ up to serial-parallel composition. If there exist elements x and y in Q such that $I_Q(x) = \{y\}$ and full subposets of Q , $P_1 = U_Q(x) \cup \{y\}$ and $P_2 = D_Q(x) \cup \{y\}$ where $e(P_1) \cdot e(D_Q(x)) = e(U_Q(x)) \cdot e(P_2)$, then $\chi(P, x, y) = \frac{1}{2}$.*

Proof. By Lemma 3.3.1, it is sufficient to show that $e(Q_{x<y}) = e(Q_{y<x})$. Because $I_Q(x) = \{y\}$, $I_{Q_{x<y}}(x) = I_{Q_{y<x}}(x) = \emptyset$. Thus, we can serially decompose both $Q_{x<y}$ and $Q_{y<x}$ about x . In particular,

$$Q_{x<y} = D_Q(x) \boxplus \{x\} \boxplus P_1$$

and

$$Q_{y<x} = P_2 \boxplus \{x\} \boxplus U_Q(x).$$

Thus

$$e(Q_{x<y}) = e(P_1) \cdot e(D_Q(x)) \text{ and } e(Q_{y<x}) = e(U_Q(x)) \cdot e(P_2).$$

By assumption these are equal, so $e(Q_{x<y}) = e(Q_{y<x})$. \square

Example 3.3.12. We illustrate Theorem 3.3.11, with the posets in Figure 3.3.13. The original poset P has a full N subposet, but by comparing x and y , we obtain posets $P_{x<y}$ and $P_{y<x}$ that can be serially decomposed about x and split perfectly.

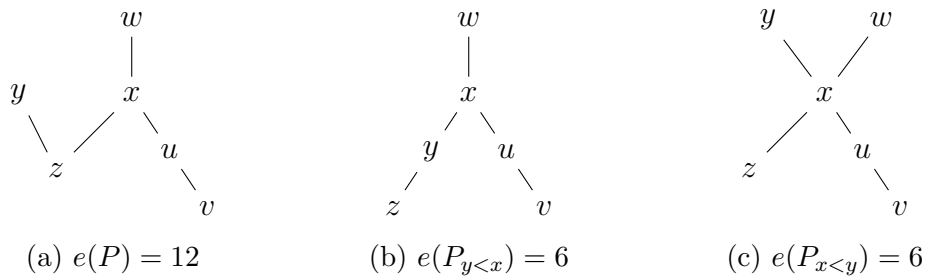


Figure 3.3.13: A perfect split from a poset with a full N .

Chapter 4

Bound Failure

Up until this point in the thesis, we have studied in great detail the structure of posets and their linear extensions. In this chapter, we shift our focus to how this structure interacts with the information-theoretic lower bound. We develop a novel framework by which to understand the tightness of the information-theoretic lower bound in relation to the structure of posets. Then, we discuss the central questions this framework helps give rise to.

4.1 Introduction to the Problem

When we defined a sorting algorithm in Chapter 1, we determined that it was a sequence $\tau(P)$ where the length $|\tau(P)|$ was the number of comparisons needed to determine a full sorting. However, this made no claims of uniqueness; for a given poset, there are likely multiple sorting sequences that are unique up to isomorphism.¹ The information-theoretic lower bound is a constraint on the length of the shortest of these sequences for a given P . We denote this bound as $\mathbb{B}(P)$.

What, then, does it mean to say that a poset is bound-failing?

Note first that the unit of the information-theoretic lower bound is in terms of the number of comparisons required to gain full knowledge of an ordering for a given set of size n . Thus, in order for a poset to meet its bound, each question in a sequence must eliminate a sufficient number of linear extensions in the resulting poset so that its bound is reduced by one. Recall that for a poset P , we denote the posets generated by comparing x and y to be $P_{x < y}$ when $x < y$ and $P_{y < x}$ when $y < x$. If the bound of P is $\mathbb{B}(P)$, and we compare x and y , a sequence that meets the bound must have $\mathbb{B}(P_{x < y}), \mathbb{B}(P_{y < x}) \leq \mathbb{B}(P) - 1$. We say that P is *bound-failing* if the minimum length sequence $|\tau(P)| > \mathbb{B}(P)$.

We can specify the exact number of linear extensions that need to be eliminated as follows. We define $e(P) = 2^i + j$, where j is an integer, $1 \leq j \leq 2^i$, and i is highest possible integer which satisfies the equation. From this, we can determine the

¹We also count algorithms that are anti-isomorphic to each other to be equivalent.

smallest fraction required to bring the bound down by one to be

$$\mathbb{S}(P) = \frac{2^i}{2^i + j}.$$

We will refer to this ratio for a given P as the *required split for P* , $\mathbb{S}(P)$. Note that if $j = 2^i$, then $\mathbb{S}(P) = \frac{1}{2}$. We say that P requires a *perfect split*, which, as its name suggests, is always the smallest (and therefore the most stringent) possible split.

A poset meets its split if there exist some pair of elements x, y such that $\chi(P, x, y) \leq \mathbb{S}(P)$; if no such comparison exists, P is *split-failing*. Because the best case split is $\frac{1}{2}$, if a poset cannot meet its split, then it requires at least $\mathbb{B} + 1$ comparisons to be fully sorted. Thus a poset that fails to meet its split will also fail to meet its bound. The converse is not true, however: a poset can meet its split while still being unable to meet its bound. Consider, for example, the completely unordered set of size 12, where $e(P) = 12!$. The first question we ask always divides $e(P)$ in half; as $\mathbb{S}(P) \geq \frac{1}{2}$ for any P , this will always meet its split. However, we know by exhaustive computer search that the unordered set of size 12 requires $\mathbb{B}(P) + 1$ comparisons [23],[17]. Hence, we must differentiate between bound failure and split failure, and disentangle the relationship between them.

4.2 Split Failure

In order to understand bound failure as a whole, we start by studying posets that fail to meet their split.

4.2.1 Small Posets that Fail to Split

Using a program that enumerates the space of sorting algorithms [1], we find that for any poset P where $|P| \leq 4$, there is at least one comparison that is at most $\mathbb{S}(P)$. The first poset that fails to meet its split occurs when $|P| = 5$, which we show below. It is the only poset of its size with this property.



Figure 4.2.1: The smallest poset that fails to meet its split.

This poset has 8 linear extensions, and thus in order to meet the bound, it would need to have a perfect split. However, no such split exists.

After $|P| > 5$, we find more than one unique split-failing poset. For $|P| = 6$, there are exactly six posets which fail to meet their split (up to anti-isomorphism.) A natural question that arises is why these particular posets cannot meet their splits.

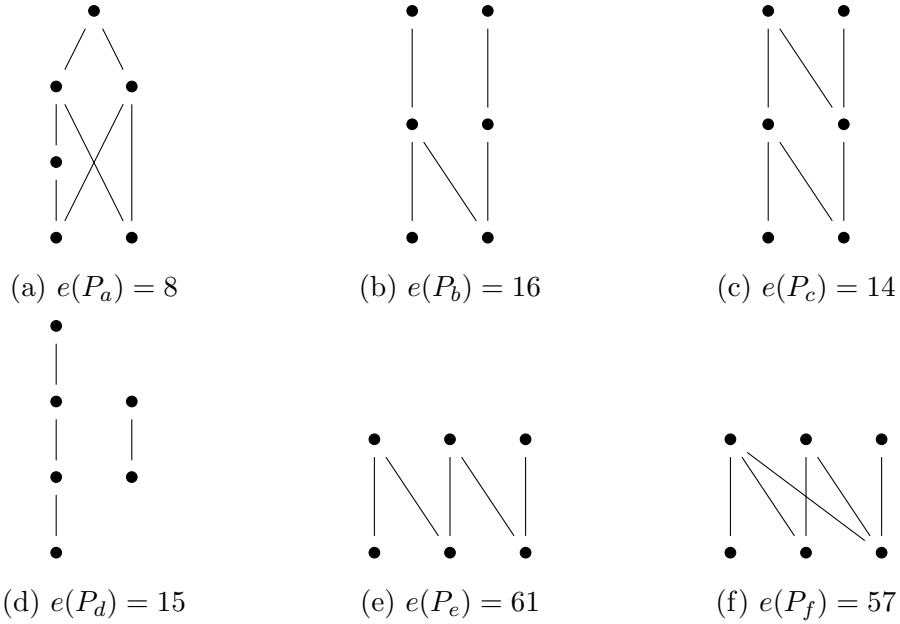


Figure 4.2.2: The set of all size 6 posets that fail to meet their split.

Numerically, these posets are unified by the proximity of the size of their set of linear extensions to a power of two, and therefore their stringent splits. Indeed, $\mathbb{S}(P_a) = \mathbb{S}(P_b) = \frac{1}{2}$, while none of the posets exceed $\mathbb{S}(P) = \frac{3}{5}$. The poset P_a is a serial composition of the 5-poset in Figure 4.2.1 and a single node, so its split failure is explained by the split-failure of the 5-poset.

Structurally, every poset has a trivial automorphism group, and all but one poset is connected. Of the connected posets, all contain an N subposet. We conjecture that this property is relevant to split failure (and thus bound failure) in general.

4.2.2 Split Requirements: Numerics

The split required of any poset P is determined by the proximity of $e(P)$ to a power of 2; the closer $e(P)$ is, the more even the split must be to bring $\mathbb{B}(P)$ down by one. Conjecture 1.1.11 discussed in Chapter 1 makes a useful threshold by which to judge how difficult $\mathbb{S}(P)$ is: because we think that any poset has a comparison $\chi(P, x, y) \leq \frac{2}{3}$, (and for certain classes of posets, we know that there exists such a split), any $\mathbb{S}(P) \geq \frac{2}{3}$ is, in a sense, easy to meet. On the other hand, for $\mathbb{S}(P) < \frac{2}{3}$, there is no guarantee that P has a pair of elements that yield such a split.

Using the same notation as Section 4.1 where $e(P) = 2^i + j$, for $i, j \in \mathbb{N}$ and $1 \leq j \leq 2^i$, we can classify $\mathbb{S}(P)$ into three categories:

1. An easy split: $\mathbb{S}(P) \geq \frac{2}{3}$, which occurs if and only if $j \leq 2^{i-1}$.
2. A hard split: $\frac{1}{2} < \mathbb{S}(P) < \frac{2}{3}$, which occurs if and only if $2^{i-1} < j < 2^i$.
3. A perfect split: $\mathbb{S}(P) = \frac{1}{2}$, which occurs if and only if $j = 2^i$.

We differentiate a hard split from a perfect split because, as Section 3.3 suggests, there are special properties that accompany having a pair of elements that perfectly divide the set of linear extensions. Moreover, asking a perfect question has no effect on the required split for subsequent posets in the sorting sequence, a fact we will return to later in the chapter.

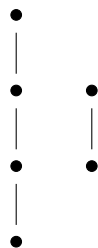
In fact, powers of 2 in $e(P)$ have no effect on $\mathbb{S}(P)$. That is, for P with $e(P)$ linear extensions, any P' with $e(P') = 2^k e(P)$ for some $k \in \mathbb{N}$, has $\mathbb{S}(P') = \mathbb{S}(P)$.

A central question this raises is identifying structural determinants that bring $e(P)$ closer or further from a power of 2 generally, and how a sequence of comparisons can together contribute to doing so. Of course, it is partially a factor of the size of P , but this certainly does not tell the whole story! We now turn to the task of teasing out other determinants.

4.2.3 Serial Composition and Splitting

In the search for structural determinants of splitting, we are faced with the unfortunate reality that very little is known about the relationship between linear extensions and general posets. One exception are series-parallel posets, which have simple formulas for determining $e(P)$ and, as we saw in Theorem 1.2.11 (due to [24]), can always meet a $\frac{2}{3}$ split. For this reason, we might suspect that split failure corresponds with being in a more “difficult” class of poset, and that we can rule out series-parallel posets as split failing.

Of course, this cannot be true for all series-parallel posets; as we saw in Section 4.2.1, the 4-2 chain poset fails to split.



However, one might think that if we impose a condition of non-trivial connectedness, we can ensure that any poset will meet its split. Intuitively, this seems plausible because a series-parallel poset can be decomposed completely using \boxplus and \oplus . We might think that by saying something about the subposets that can be obtained by \boxplus and \oplus decompositions, we could say something about the relationship between P and $\mathbb{S}(P)$ as a whole. Proving this would be equivalent to saying that any non-splitting, non-trivial connected poset must contain a full N subposet by Theorem 1.2.10.

One seemingly viable proof technique would be to prove by induction on the size of the poset that any connected series-parallel poset meets its split. The base case is trivial, as all sufficiently small posets ($|P| \leq 4$) meet their split. After making the inductive hypothesis, we would make a serial decomposition of P , leaving us with two posets which by assumption meet their splits. We would therefore need to prove that the serial composition of two split meeting posets meets its split.

However, there are two problems with this approach, only one of which is remediable. First: every serial parallel poset (aside from a single, totally ordered chain), at some point in its decomposition must become a parallel composition (and thus be disconnected). We would therefore have to impose a condition on the permissible types of disconnectedness in a poset, which might involve a maximal height. It is not worth even attempting to determine these conditions, however, because this approach has a deeper problem: serial composition does not necessarily preserve split-meeting. In fact, it does not even preserve bound.

Serial Composition and Bounds

As serial compositions increase the count of linear extensions multiplicatively, we might think that the information-theoretic lower bound of a serially composed poset is the sum of its parts. As it turns out, this is only sometimes true; it is possible that in making a serial composition between two posets, the bound of the whole is one less than the sum of its parts.

Proposition 4.2.4. *Let P_1, P_2 be posets with $e(P_1) = 2^i + j$ and $e(P_2) = 2^k + l$ for $1 \leq j \leq 2^i$, $1 \leq l \leq 2^k$. Then*

$$\mathbb{B}(P_1 \boxplus P_2) = \begin{cases} \mathbb{B}(P_1) + \mathbb{B}(P_2) - 1 & \text{if } 2^i l + 2^k j + jl \leq 2^{i+k} \\ \mathbb{B}(P_1) + \mathbb{B}(P_2) & \text{if } 2^i l + 2^k j + jl > 2^{i+k}. \end{cases}$$

Proof. By above, $\mathbb{B}(P_1) = i + 1$, $\mathbb{B}(P_2) = k + 1$. So

$$e(P_1 \boxplus P_2) = (2^i + j)(2^k + l) = 2^{i+k} + 2^i l + 2^k j + jl \leq 4(2^{i+k}).$$

If $2^{i+k} < 2^i l + 2^k j + jl$ then

$$2^{i+k+1} < 2^{i+k} + 2^i l + 2^k j + jl \leq 2^{i+k+2}.$$

Thus $\mathbb{B}(P_1 \boxplus P_2) = i + k + 2 = \mathbb{B}(P_1) + \mathbb{B}(P_2)$. Now if

$$2^i l + 2^k j + jl \leq 2^{i+k}$$

then $e(P) \leq 2^{i+k+1}$, so $\mathbb{B}(P_1 \boxplus P_2) = i + k + 1 = \mathbb{B}(P_1) + \mathbb{B}(P_2) - 1$. \square

From Proposition 4.2.4 it becomes clear that when we analyze serial composition, we must divide our analysis into two cases; the normal case, in which the bound is the sum of its bounds, and the messy case, in which the bound decreases by one.

The Normal Case

In the normal case, things behave exactly as we want them to; for any two posets that we serially compose, the split required of their composition is greater than the split needed by either component. Thus if two such posets meet their split, their serial composition must also meet its split.

Proposition 4.2.5. *If $P = P_1 \boxplus P_2$ and $\mathbb{B}(P) = \mathbb{B}(P_1) + \mathbb{B}(P_2)$, then $\mathbb{S}(P) \geq \mathbb{S}(P_1), \mathbb{S}(P_2)$.*

Proof. Let $e(P_1) = 2^i + j$ and $e(P_2) = 2^k + l$ for $1 \leq j \leq 2^i, 1 \leq l \leq 2^k$. By assumption $\mathbb{B}(P) = 2 + i + k$, and therefore

$$\mathbb{S}(P) = \frac{2^{i+k+1}}{(2^i + j)(2^k + l)} \text{ and } \mathbb{S}(P_1) = \frac{2^i}{2^i + j}.$$

By our assumption that $l \leq 2^k$,

$$2^{k+1} \geq 2^k + l,$$

thus

$$2^{i+k+1}(2^i + j) \geq 2^i(2^i + j)(2^k + l)$$

so

$$\frac{2^{i+k+1}}{(2^i + j)(2^k + l)} \geq \frac{2^i}{2^i + j}.$$

Therefore $\mathbb{S}(P) \geq \mathbb{S}(P_1)$. By symmetry, $\mathbb{S}(P) \geq \mathbb{S}(P_2)$. \square

It follows immediately that if either P_1 or P_2 meet their splits, P must also.

The Messy Case

As the name suggests, things do not work out nearly so well in our messy case. Recall by Proposition 4.2.4 that for $P = P_1 \boxplus P_2$, if $\mathbb{B}(P) = \mathbb{B}(P_1) + \mathbb{B}(P_2) - 1$ then for $e(P_1) = 2^i + j$ and $e(P_2) = 2^k + i$, we have that $2^{i+j} \geq 2^i l + 2^k j + lk$. In this case, the split of the serial composition of P_1 and P_2 is more stringent than the split of either component.

Proposition 4.2.6. *Let P be a poset that can be decomposed serially into two posets $P_1 \boxplus P_2$, where $\mathbb{B}(P) = \mathbb{B}(P_1) + \mathbb{B}(P_2) - 1$. Then $\mathbb{S}(P) < \mathbb{S}(P_1), \mathbb{S}(P_2)$.*

Proof. By assumption we have that $\mathbb{B}(P) = i + k + 1$, $\mathbb{B}(P_1) = i + 1$, and $\mathbb{B}(P_2) = k + 1$. Therefore,

$$\mathbb{S}(P) = \frac{2^{i+k}}{(2^i + j)(2^k + l)} = \left(\frac{2^i}{2^i + j} \right) \left(\frac{2^k}{2^k + l} \right) = \mathbb{S}(P_1) \mathbb{S}(P_2).$$

Clearly, $0 < \mathbb{S}(P_1), \mathbb{S}(P_2) < 1$, and therefore $\mathbb{S}(P) < \mathbb{S}(P_1), \mathbb{S}(P_2)$. \square

Proposition 4.2.6 thus gives us a potential method of split failure that is hardwired into serial composition, and calls into question the idea of a connected split failing always coming from a poset with a full N subposet.

One potential (and unsuccessful) way around this would be if the split of $P_1 \boxplus P_2$ (as defined above) was always at least $\frac{2}{3}$, in which case any series parallel poset would be able to meet that split by Theorem 1.2.11. However, this is not the case; $\mathbb{S}(P)$ can easily fall below $\frac{2}{3}$ when $2^{i+k-1} < 2^i l + 2^k j + lj \leq 2^{i+k}$.

4.3 Bound Failure

We now turn to the larger problem of bound-failure.

4.3.1 Inheriting Bound Failure

It is clear that bound failure is related to split failure; we now formalize this relationship.

A split failing poset does not exist in isolation, and to understand it, we must consider not just the structure of the poset that fails to meet its split, but also the structure of the comparisons that preceded it. Rather than looking at individual partial orders as the source of bound failure, we now try to integrate them into bound failing sequence as a whole.

The first important thing to note is that not every comparison in a split-meeting poset is a “good” question. That is, even though there exist some $x, y \in P$ such that $\chi(P, x, y) \leq \mathbb{S}(P)$, there may be another pair of elements z, w such that $\mathbb{S}(P) < \chi(P, z, w)$. In this case, we say that z and w yield a *bad split* while x and y yield a *good split*.

This insight has useful implications for analyzing posets that are bound-failing but not split-failing. We say such posets *inherit* bound failure. Formally, a poset P *inherits* bound failure if there exist a pair of elements $x||y \in P$ such that $\chi(P, x, y) \leq \mathbb{S}(P)$ but the minimum length sorting sequence is at least $\mathbb{B}(P) + 1$.

With this terminology in hand, we can give a definition of bound-failure that incorporates both bound inheritance and split-failure.

Definition 4.3.1. A poset P is bound failing if for every sorting sequence of P , $\tau(P) = P, P_1, P_2, \dots, P_n$ there are at least two consecutive terms P_m, P_{m+1} in $\tau(P)$, for $1 \leq m < n$ such that $\mathbb{B}(P_m) = \mathbb{B}(P_{m+1})$.

To see why this is an apt definition of bound failure, we enumerate the possible cases for a bound failing poset P in terms of the comparisons that can be made in it. Without loss of generality, let $e(P_{x<y}) \geq e(P_{y<x})$. There are three cases:

Case I. *P is split failing.* Then for any $x||y \in P$, $\mathbb{B}(P) = \mathbb{B}(P_{x<y})$.

Case II. *The child of P, P_{x<y}, is made from a bad split.* Then this is equivalent to Case I. Note that $P_{x<y}$ need not be bound failing; if it meets its bound, then there is some sequence of comparisons that sort $P_{x<y}$ in exactly $\mathbb{B}(P_{x<y})$ comparisons. This tells us that P can be sorted in $\mathbb{B}(P_{x<y}) + 1$ comparisons, because we have one additional comparison between x and y . As we know P is bound failing, $\mathbb{B}(P) < \mathbb{B}(P_{x<y}) + 1$. So $\mathbb{B}(P) = \mathbb{B}(P_{x<y})$.

Case III. *The child of P, P_{x<y} is generated by a good split.* Then $\mathbb{B}(P_{x<y}) = \mathbb{B}(P) - 1$. We know that $P_{x<y}$ must also be bound failing because otherwise P would not be bound failing using the same sorting sequence that sorts $P_{x<y}$ in $\mathbb{B}(P_{x<y})$ comparisons.

Now consider the sorting sequences beginning at $P_{x<y}$. If $P_{x<y}$ is split failing, then Case I applies. For any question that yields a bad split, Case II applies. Thus assume that $P_{x<y}$ meets its split, and follow any sequence generated by good splits. We claim that every such sequence must eventually meet a bad split, or $P_{x<y}$ would meet its bound, and therefore so would P . The point in which the sequence meets a bad split must then reduce to either Case I or II.

Definition 4.3.1 thus shifts our focus. It tells us that while split failure is a necessary component of bound failure, we must also consider the split failure which occurs by way of bad splits in split meeting posets. Through this lens, rather than thinking of split-failing posets as the root cause of bound failure, they can be understood as the end result of an algorithm that procrastinates by prolonging asking a bad question for as long as possible.

The natural next question becomes why certain posets are forced to have sorting sequences that contain a bad split, while others are not.

4.3.2 The Sequence of Required Splits

While there are no doubt structural explanations as to why particular sorting sequences are forced to ask bad questions, we can learn a substantial amount about how the bound interacts with sorting algorithms by looking at how the required splits change in a given sorting sequence.

For any sorting sequence $\tau = P_0, P_1, \dots, P_n$, we can form an accompanying sequence $\Sigma(\tau) = \mathbb{S}(P_0), \mathbb{S}(P_1), \dots, \mathbb{S}(P_n)$. We might wonder how $\Sigma(\tau)$ behaves for a bound meeting sequence τ .

Imperfect Questions

As we have discussed, a perfect split has no effect on the required split. We now show that any imperfect good split strictly decreases the required split for the subsequent term in τ .

Theorem 4.3.2. *Let $s \in \mathbb{R}$, where $\frac{1}{2} < s < 1$. Let P be a poset with $x, y \in P$ such that $\chi(P, x, y) = s \leq \mathbb{S}(P)$. Then $\mathbb{S}(P_{x<y}) < \mathbb{S}(P)$.*

Proof. Let $e(P) = 2^i + j$ where $1 \leq j \leq 2^i$. With $\chi(P, x, y) = s$ as above, we have that $e(P_{x<y}) = s(2^i + j)$. Because $\chi(P, x, y) \leq \mathbb{S}(P)$ and $s > \frac{1}{2}$, we know that $2^{i-1} < e(P_{x<y}) \leq 2^i$, so

$$\mathbb{S}(P_{x<y}) = \frac{2^{i-1}}{s(2^i + j)}.$$

Because

$$s > \frac{1}{2} = \frac{(2^i + j)(2^{i-1})}{(2^i + j)(2^i)},$$

we have

$$s(2^i)(2^i + j) > (2^i + j)(2^{i-1})$$

and therefore

$$\frac{2^i}{2^i + j} > \frac{2^{i-1}}{s(2^i + j)},$$

which is precisely $\mathbb{S}(P) > \mathbb{S}(P_{x < y})$. \square

While the proof is quite elementary, Theorem 4.3.2 tells us something significant about how a sorting sequence behaves over time. In fact, it tells us that each imperfect question asked in a sorting sequence makes it more difficult for subsequent questions to meet their splits. As we might expect, a worse (but still split meeting) split magnifies this effect.

Given two splits $s < s'$ for the same poset,

$$\frac{2^{i-1}}{s(2^i + j)} > \frac{2^{i-1}}{s'(2^i + j)}.$$

So a worse split leads to a strictly smaller required split for the resulting sequence.

Theorem 4.3.2 also tells us something about the asymptotic behavior of the required split for a sequence of split-meeting posets. For a sorting sequence $\tau(X) = P_0, P_1, \dots, P_n$, if every comparison is split meeting, then $\mathbb{S}(P_i) \geq \mathbb{S}(P_{i+1})$, with equality only when the question asked to get from P_i to P_{i+1} is perfect.

Thus $\Sigma(\tau) = \mathbb{S}(P_0), \mathbb{S}(P_1), \dots, \mathbb{S}(P_n)$ is a bounded, monotonic sequence.

The $\frac{2}{3}$ Split

From Theorem 4.3.2, it is apparent that any imperfect split can have ramifications on the ability of a sorting sequence to meet its bound. We might be curious, therefore, about the effect of a $\frac{2}{3}$ split on $\Sigma(\tau)$, given that by the $\frac{2}{3}$ Conjecture (Conjecture 1.1.11), a $\frac{2}{3}$ split is, in theory, the best general claim that we can make about the split of an arbitrary poset. It turns out that if a sorting sequence is to meet its bound, it can ask a maximum of two such questions!

Theorem 4.3.3. *Any bound meeting sorting sequence can ask at most two questions yielding a $\frac{2}{3}$ split.*

Proof. Let P be an arbitrary poset with $e(P) = 2^i + j$, so $\mathbb{B}(P) = i + 1$. We make two comparisons, q_1 and q_2 both of which yield a $\frac{2}{3}$ split. Let P_1 be the resulting poset after asking q_1 of P and P_2 be the resulting poset after asking q_2 of P_1 . By Theorem 4.3.2, we may assume without loss of generality that q_1 and q_2 are asked consecutively, as that is the best case scenario with respect to the values of $\mathbb{S}(P_1), \mathbb{S}(P_2)$.

1. *Asking q_1 :* In order to make this split, $j \leq 2^{i-1}$. If the split is made, $\mathbb{B}(P_1) = i$,

$$e(P_1) = \frac{2(2^i + j)}{3}$$

and therefore the split of P_1 is

$$\mathbb{S}(P_1) = \frac{3(2^{i-1})}{2(2^i + j)} = \frac{3(2^{i-2})}{2^i + j} < \frac{2^i}{2^i + j} = \mathbb{S}(P).$$

For what values of j is $\mathbb{S}(P_1) < \frac{2}{3}$? This is satisfied when

$$\frac{3(2^{i-2})}{2^i + j} < \frac{2}{3}$$

so

$$9(2^{i-2}) = (1 + 2^3)(2^{i-2}) = 2^{i-2} + 2^{i+1} < 2^{i+1} + 2j$$

which is equal to

$$2^{i-3} < j.$$

Thus for $j \leq 2^{i-3}$, we have that $\mathbb{S}(P_1) \geq \frac{2}{3}$, which means that when j is in this range, we can ask a second $\frac{2}{3}$ -split question and meet $\mathbb{S}(P_1)$.

2. *Asking q_2* : Assuming that P_1 met its split with q_2 , $\mathbb{B}(P_2) = i - 1$. When we ask q_2 , we get

$$e(P_2) = \frac{4(2^i + j)}{9}.$$

This gives us a split of

$$\mathbb{S}(P_2) = \frac{9(2^{i-2})}{2^{i+2} + 4j} < \frac{3(2^{i-2})}{2^i + j} = \mathbb{S}(P_1).$$

For what values of j is $\mathbb{S}(P_2) < \frac{2}{3}$? We require

$$\mathbb{S}(P_2) = \frac{9(2^{i-2})}{2^{i+2} + 4j} < \frac{2}{3}$$

giving us

$$27(2^{i-2}) < 2(2^{i+2} + 4j) = 2^{i+3} + 8j$$

which is equivalent to

$$27(2^{i-2}) - 2^{i+3} = 2^{i-2}(27 - 2^5) = 2^{i-2}(27 - 32) < 8j$$

By assumption j is non-negative, so for every value of j , $\mathbb{S}(P_2) < \frac{2}{3}$. So there is no way to ask a third $\frac{2}{3}$ question and meet the split.

□

By Theorem 4.3.3, we deduce two important facts. First, asking a $\frac{2}{3}$ split question significantly reduces the required split for resulting posets in τ . In order to make one $\frac{2}{3}$ split, we require that $j \leq 2^{i-1}$; in order to have two split-meeting $\frac{2}{3}$ questions, we need $j \leq 2^{i-3}$. Thus even asking one $\frac{2}{3}$ question will put stress on the rest of the sorting sequence. A good example of this phenomenon occurs as early as $|P| = 6$, which we show in Figure 4.3.4. Although the unordered set for $|P| = 6$ can be sorted in $\mathbb{B}(P)$ comparisons, asking the first possible $\frac{2}{3}$ question significantly reduces the required split, forcing a series of perfect splits and eventually, split-failing posets.

$$P_0 : e(P_0) = 360. \mathbb{S}(P_0) = \frac{256}{360} = .71$$

$$\text{Split: } \frac{2}{3}$$



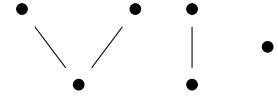
$$P_1 : e(P_1) = 240. \mathbb{S}(P_1) = \frac{8}{15} = .53$$

$$\text{Split: } \frac{1}{2}$$



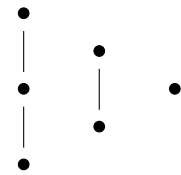
$$P_2 : e(P_2) = 120. \mathbb{S}(P_2) = \frac{8}{15} = .53$$

$$\text{Split: } \frac{1}{2}$$



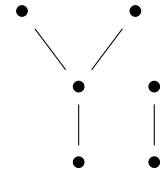
$$P_3 : e(P_3) = 60. \mathbb{S}(P_3) = \frac{8}{15} = .53$$

$$\text{Split: } \frac{1}{2}$$



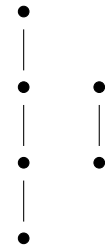
$$P_4 : e(P_4) = 30. \mathbb{S}(P_4) = \frac{8}{15} = .53$$

Split from P_4 : $\frac{1}{2}$. This yields a non-splitting poset



$$P_{5a} : e(P_{5a}) = 15. \mathbb{S}(P_{5a}) = \frac{8}{15} = .53$$

Alternative split from P_4 : $\frac{8}{15}$, also yielding a non-splitting poset



$$P_{5b} : e(P_{5b}) = 16. \mathbb{S}(P_{5b}) = \frac{1}{2} = .5$$

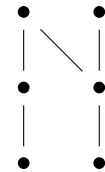


Figure 4.3.4: The ramifications of a $\frac{2}{3}$ split for a poset of size 6.

Second, Theorem 4.3.3 tells us that Conjecture 1.1.11 is quite incompatible with the information-theoretic lower bound. It suggests that the lower bound measure we have for the length of a sorting sequence does not nicely incorporate the bound on

splits that we have at the level of individual posets. While we (maybe) can divide any poset into thirds, this does nothing to guarantee that a sorting sequence as a whole can meet its split. In fact, asking a $\frac{2}{3}$ question consistently is a strategy that is guaranteed to fail the bound! This fact suggests that there is a disconnect between the structure of a poset and the numerical proof that the information-theoretic lower bound derives from.

4.3.3 Bound Failure in the Unordered Set

In Chapter 1, we stated that the information-theoretic lower bound was tight for sets of size $n \leq 11$, $n = 20, 21$ but not for $12 \leq n \leq 19$, $n = 22$. We now return to this strange result.

One thing that the previous discussion illuminates is that in looking at the unordered sets that are bound failing, we should be asking which paths are rendered ineffective immediately (or close to the beginning of an algorithm). A natural place to begin is by determining the splits initially required for an unordered set.

Below, we show the required splits for $5 \leq n \leq 22$, with the bound failing set sizes in bold. We also list the split *after* a $\frac{2}{3}$ split, which as we can see, is far smaller than the initial split.²

n	Split Required	Meet $\frac{2}{3}$	Split after $\frac{2}{3}$
5	.53	no	.66
6	.71	yes	.53
7	.81	yes	.61
8	.81	yes	.61
9	.72	yes	.54
10	.58	no	.87
11	.84	yes	.63
12	.56	no	.84
13	.69	yes	.52
14	.79	yes	.59
15	.84	yes	.63
16	.84	yes	.63
17	.79	yes	.59
18	.70	yes	.53
19	.59	no	.89
20	.95	yes	.71
21	.72	yes	.54
22	.53	no	.79

Table 4.1: The splits required for an unsorted set of size n

Clearly, this is a coarse metric by which to judge bound failure; we can see that

²This is true only of posets for which $\frac{2}{3}$ is a good split. For posets that already have $\mathbb{S}(P) < \frac{2}{3}$, this does not apply.

for $n = 12$, the first unordered poset that fails the bound, the split is highly stringent. However, the split for $n = 5$ is even lower, and it meets its bound!

The Greedy Approach and $n = 12$

The sorting algorithm for $n = 5$ is narrowly bound meeting, despite having an extremely stringent split. That is, for the first four comparisons of the sorting algorithm, there is only one good split (up to anti-isomorphism) that can be made. We call this approach the *greedy approach* because it asks the comparison that yields the best split. The greedy approach is certainly not always optimal. In 1974, Warren published a paper pointing out this fact ([22]), and we know by Peczarski that for $n = 16$, asking a series of perfect questions does not yield even an algorithm that makes $\mathbb{B}(P) + 1$ comparisons [18]. We show the beginning of the greedy algorithm for $n = 5$ in Figure 4.3.5. Note that the first three questions yield a perfect split.

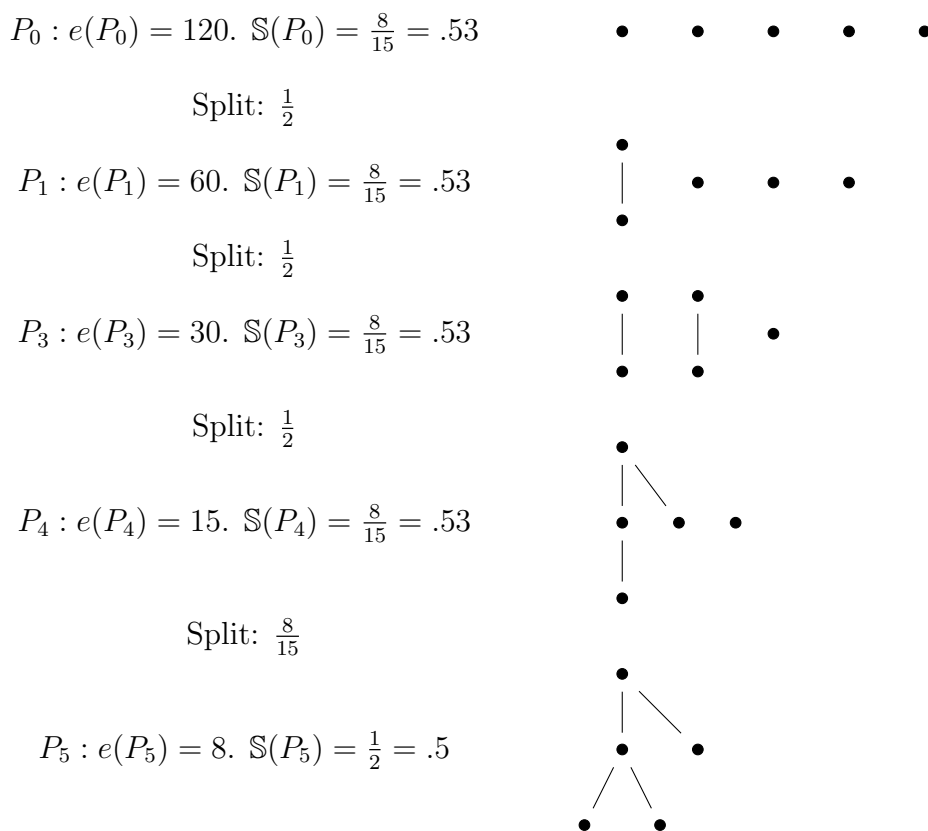


Figure 4.3.5: The beginning of the greedy algorithm for $n = 5$.

We might wonder how this approach fares for sets of other sizes. How many perfect questions can we initially ask?

For any unordered set of size n , there are $n!$ linear extensions. By Legendre's formula, the largest $m \in \mathbb{N}$ such that 2^m divides $n!$, denoted $\nu_2(n!)$, is given by

$$\nu_2(n!) = \sum_{i \geq 1} \left\lfloor \frac{n}{2^i} \right\rfloor,$$

where the sum is necessarily finite, as n is fixed. It turns out that by Theorem 3.3.2 we can guarantee just as many perfect splits as $\nu_2(n!)$.

Proposition 4.3.6. *For any unordered set of size n , there is a sorting sequence that yields $\nu_2(n!)$ perfect consecutive splits.*

Proof. Let X be the underlying set of the sorting sequence, where $|X| = n$. We use Theorem 3.3.2, which shows that if there is a $\pi \in \text{aut}(P)$ that transposes two elements, then comparing them gives a perfect split. We prove by induction on the i in the summation

$$\nu_2(n!) = \sum_{i \geq 1} \left\lfloor \frac{n}{2^i} \right\rfloor$$

that by creating parallel compositions of isomorphic subposets, we can ask $\nu_2(n!)$ perfect questions.

For the base case, we group the discrete poset into $\left\lfloor \frac{n}{2} \right\rfloor$ pairs. Clearly, for any two isolated elements $x, y \in X$, there must be some $\pi \in \text{aut}(P)$ that transposes x and y , and so comparing them gives a perfect split. We make $\left\lfloor \frac{n}{2} \right\rfloor$ such comparisons.

For the induction step, assume that we have asked $j = \left\lfloor \frac{n}{2^{i-1}} \right\rfloor$ questions with perfect splits, giving us a poset P ,

$$P = P_1 \oplus \dots \oplus P_j \oplus Q,$$

where $Q = P - (P_1 \oplus \dots \oplus P_j)$ and for any P_k, P_l , $1 \leq k, l \leq j$, we have $P_k \cong P_l$. Group P_1, P_2, \dots, P_j into pairs, of which there will be precisely

$$m = \left\lfloor \frac{j}{2} \right\rfloor = \left\lfloor \frac{n}{2^i} \right\rfloor.$$

We make comparisons as follows. For any k , $1 \leq k \leq j$, fix an isomorphism $\phi_k : P_1 \rightarrow P_k$. Let $x \in P_1$. For each pair of subposets, P_k, P_l , compare $\phi_k(x)$ with $\phi_l(x)$. Because P_k and P_l are disjoint and isomorphic, there must be some $\sigma \in \text{aut}(P)$ such that $\sigma(P_k) = P_l$ and $\sigma(P_l) = P_k$, where σ transposes $\phi_k(x)$ and $\phi_l(x)$. By Theorem 3.3.2 their comparison will give a perfect split. We are thus left with

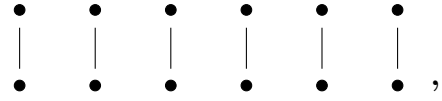
$$P' = P'_1 \oplus \dots \oplus P'_m \oplus Q',$$

where $Q' = P' - (P'_1 \oplus \dots \oplus P'_m)$ and for any P'_r, P'_s , $1 \leq r, s \leq m$, we have $P'_r \cong P'_s$.

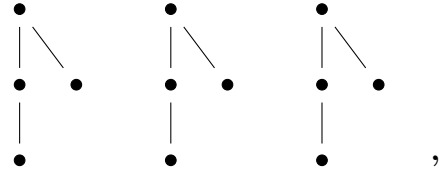
Let P_ν be the poset we obtain after asking $\nu_2(n!)$ perfect questions. Then $2 \nmid e(P_\nu)$, and thus it is impossible to ask an additional perfect question of P_ν . \square

We show such a sorting sequence for $n = 12$, where $\nu_2(12!) = 10$.

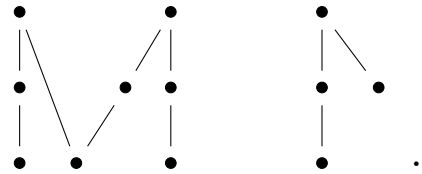
The first six perfect questions give



while the next three perfect questions give



and the last perfect question leaves us with



Note that after we ask the final perfect question, we are left with a poset that not only has a trivial automorphism group, but one that is structurally quite complicated. While the last perfect question shown above is not the only perfect question that can be asked of that poset (there are four, up to isomorphism), any of these questions will leave us with a poset that has such an automorphism group. It is also worth noting that Proposition 4.3.6 does not tell us that the sorting sequence on a poset can have only $\nu_2(n!)$ perfect questions, but rather that by asking $\nu_2(n!)$ perfect questions initially, we must ask at least one imperfect question in order to be able to have a perfect split in subsequent posets in the sorting sequence.

Can this help explain why the unordered set of size 12 is bound-failing? While it is not the full explanation, it certainly lends insight to the problem. In order for a poset to become sorted, it must first become completely connected. Interestingly, by careful calculation, for $n = 12$ the greedy strategy is the only method by which we can obtain a parallel composition of two connected subposets without making a bad split.

Conclusion

This thesis began with a question about how the structure of partial preferences interacts with the process of revealing them—a problem which is classically framed in terms of posets and the information-theoretic lower bound for sorting. What we have found, however, is that there seems to be a disparity between the structure of a partial order and the demands made by the bound in sorting it. We have tried to draw a dichotomy between, on the one hand, the way we understand posets structurally—the splits they can make, the way their symmetries interact with their linear extensions, the way (some of them) can be composed and decomposed—and on the other, the way the information-theoretic lower bound for sorting operates on a sequence of posets. If anything, the work in this thesis has shown that the structure of partial orders is quite nuanced, and that perhaps the information-theoretic lower bound is too blunt a weapon to use against it.

Works Cited

- [1] Sarah Brauner. How good is the information theoretic bound for sorting? REU Write-up, Santa Fe Insitute, 2015.
- [2] Graham Brightwell. Balanced pairs in partial orders. *Discrete Mathematics*, 201(1):25–52, 1999.
- [3] Jacques Crémer and Michael H. Riordan. A sequential solution to the public goods problem. *Econometrica*, 53(1):77–84, 1985.
- [4] Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166, 1950.
- [5] John D Dixon and Brian Mortimer. *Permutation groups*, volume 163. Springer Science & Business Media, 1996.
- [6] Michael L Fredman. How good is the information theory bound in sorting? *Theoretical Computer Science*, 1(4):355–361, 1976.
- [7] Bernhard Ganter, Gerhart Häfner, and Werner Poguntke. On linear extensions of ordered sets with a symmetry. *Discrete Mathematics*, 63(2):153–156, 1987.
- [8] Jerry Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45(2):427–438, 1977.
- [9] Satoshi Hanamura and Shigeki Iwata. Posets with seven linear extensions sortable by three comparisons. *Information Processing Letters*, 111(8):365–369, 2011.
- [10] David Kelly and William T Trotter Jr. *Dimension theory for ordered sets*. Springer, 1982.
- [11] SS Kislitsyn. A finite partially ordered set and its corresponding set of permutations. *Mathematical Notes of the Academy of Sciences of the USSR*, 4(5):798–801, 1968.
- [12] Donald Ervin Knuth. *The art of computer programming: sorting and searching*, volume 3. Pearson Education, 1998.

-
- [13] Nathan Linial. The information-theoretic bound is good for merging. *SIAM J. Comput.*, 13(4):795–801, 1984.
- [14] Dennis C Mueller. Public choice: An introduction. In *Readings in Public Choice and Constitutional Political Economy*, pages 31–46. Springer, 2008.
- [15] Joseph Neggers and Hee Sik Kim. *Basic posets*. World Scientific, 1998.
- [16] Marcin Peczarski. New results in minimum-comparison sorting. *Algorithmica*, 40(2):133–145, 2004.
- [17] Marcin Peczarski. The ford–johnson algorithm still unbeaten for less than 47 elements. *Information processing letters*, 101(3):126–128, 2007.
- [18] Marcin Peczarski. Towards optimal sorting of 16 elements. *arXiv preprint arXiv:1108.0866*, 2011.
- [19] Ivan Rival. Stories about order and the letter n (en). *Combinatorics and ordered sets*, pages 263–285, 1986.
- [20] Geoff Smith and Olga Tabachnikova. *Topics in group theory*. Springer Science & Business Media, 2012.
- [21] Jacobo Valdes, Robert E. Tarjan, and Eugene L. Lawler. The recognition of series parallel digraphs. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing, STOC '79*, pages 1–12, New York, NY, USA, 1979. ACM.
- [22] Henry S Warren. Minimal comparison sorting by choosing most efficient comparisons. *Information Processing Letters*, 2(5):129–130, 1974.
- [23] Mark B Wells. Applications of a language for computing in combinatorics. Technical report, Los Alamos Scientific Lab., Univ. of California, N. Mex., 1964.
- [24] Imed Zaguia. The 1/3-2/3 conjecture for n -free ordered sets. *arXiv preprint arXiv:1107.5626*, 2011.