On Counting Posets and the Structure of the Poset of Posets

By

Joseph C. Culberson
University of Alberta
Edmonton, Alberta, Canada T6G 2H1

and

Gregory J. E. Rawlins
Indiana University
Bloomington, IN 47405

TECHNICAL REPORT NO. 266

On Counting Posets and the Structure of the Poset of Posets

by

Joseph C. Culberson and Gregory J. E. Rawlins

December 1988

On Counting Posets and the Structure of the Poset of Posets

Joseph C. Culberson * Gregory J. E. Rawlins †

December 15, 1988

Abstract

In this paper we report on the success of a new technique for computing the number of unlabeled partial orders on n elements based on the partial order of partial orders ordered by containment. In addition to the number of partial orders, we obtain complete enumerations of the number of partial orders on n elements with r relations for $n \leq 11$, where r takes on all possible values. We point out some interesting sequences that arise in these tables.

1 Introduction

In the following sections of this paper we describe an algorithm used to generate a computer enumeration of the number of unlabeled posets on n elements and r relations for $0 \leq n \leq 11$ and $0 \leq r \leq \binom{n-1}{2}$. This algorithm is dependent on some simple observations about the structure of the poset of posets on n elements, which we designate $\mathcal{P}(n)$.

The problem of obtaining a closed formula for such an enumeration is still open, and so far there does not appear to exist even a simple method of computing the number of posets, either labeled or unlabeled, short of actually generating them. For this reason, enumeration for small n has been carried out using a computer to actually generate all posets and count them. Previous attempts have been made at such enumerations. Evans, Harary and Lynn [?] used a computer to enumerate the number of labeled posets and the number of labeled topologies for $n \leq 7$. For unlabeled posets, Mohring [?] generated results for $n \leq 10$, based on the identification of comparability graphs selected from the Read-Wormald database of graphs [?]. (His results do not agree with ours for $n = 10$).

2 Poset Definitions

A *partial order* $>$ is an irreflexive and transitive (and hence asymmetric) binary relation. A *partially ordered set*, or *poset* is a structure $(\mathcal{A}, >)$ with a partial order $>$ defined on the elements of $\mathcal{A}$. For convenience we refer to the structure $(\mathcal{A}, >)$ as just $\mathcal{A}$. We use the calligraphic letters $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ to stand for posets. All posets discussed are assumed to be *finite* and their elements are assumed to be chosen from a totally ordered set, that is, if $x \neq y$ then either $x > y$ or $y > x$.

*Supported by Natural Sciences and Engineering Research Council Grant No. OGP8053. Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, T6G 2H1. email: joe@alberta.uucp

†Department of Computer Science, Indiana University, 101 Lindley Hall, Bloomington, Indiana 47405, USA. email: rawlins@iuvax.cs.indiana.edu

x and y are said to be *related* in $\mathcal{A}$ if either $x > y$ or $y > x$ is in $\mathcal{A}$, otherwise x and y are *unrelated* in $\mathcal{A}$. If x and y are unrelated then we write $x \parallel y$. We use $E(\mathcal{A})$ to represent the set of relations in $\mathcal{A}$. $x \in \mathcal{A}$ is said to be a *singleton* if it is unrelated to every other element in $\mathcal{A}$. The set of singletons of $\mathcal{A}$ is denoted *singletons*$(\mathcal{A})$. $\mathcal{A}$ is said to be in *reduced form* if it contains no singletons. Two elements form a *pair* if they are only related to each other. The *dual* of $\mathcal{A}$ is the poset $\mathcal{A}^*$ for which $x > y$ in $\mathcal{A}$ if and only if $y > x$ in $\mathcal{A}^*$.

x *covers* y ($x \succ y$) in $\mathcal{A}$ if $x > y$ in $\mathcal{A}$ and $x > z, z > y$ in $\mathcal{A}$ implies that $z = x$ or $z = y$.

A *chain* (*antichain*) is a poset in which all elements are pairwise related (unrelated). The chain and antichain on n elements is denoted by $\mathcal{R}_n$ and $\mathcal{U}_n$, respectively.

Let $\mathcal{P}(n)$ be the set of all posets on n elements. Given two posets $\mathcal{A}$ and $\mathcal{B}$ in $\mathcal{P}(n)$ we say that $\mathcal{A}$ *contains* $\mathcal{B}$ ($\mathcal{A} \geq \mathcal{B}$) if there is an order-preserving injection from $\mathcal{B}$ into $\mathcal{A}$. $\mathcal{P}(n)$ forms a poset under containment with $\mathcal{R}_n$ as unique maximal element and $\mathcal{U}_n$ as unique minimal element. If $\mathcal{A} \geq \mathcal{B}$ in $\mathcal{P}(n)$ then we say that $\mathcal{B}$ is a *subposet* of $\mathcal{A}$. If there is an order-preserving bijection between $\mathcal{A}$ and $\mathcal{B}$ then we say that $\mathcal{A}$ and $\mathcal{B}$ are *isomorphic* ($\mathcal{A} \simeq \mathcal{B}$). If we wish to speak of a specific bijection f we use the relation $\mathcal{A} \simeq_f \mathcal{B}$. We use the notation $\mathcal{A} \leq \mathcal{B}$ to mean $\mathcal{B} \geq \mathcal{A}$ and the notation $\mathcal{A} < \mathcal{B}$ to mean that $\mathcal{A} \leq \mathcal{B}$ and $\mathcal{A} \not\simeq \mathcal{B}$. Since $\mathcal{P}(n)$ is a poset, we freely use the notation developed previously. For example, we say that $\mathcal{A}$ *covers* $\mathcal{B}$ ($\mathcal{A} \succ \mathcal{B}$) in $\mathcal{P}(n)$ if $\mathcal{A} \geq \mathcal{B}$ and $\mathcal{A} \geq \mathcal{C}, \mathcal{C} \geq \mathcal{B}$ in $\mathcal{P}(n)$ implies that $\mathcal{C} \simeq \mathcal{A}$ or $\mathcal{C} \simeq \mathcal{B}$.

A poset $\mathcal{A}$ is *graded* if there exists a function $f : \mathcal{A} \to \mathbf{N}$ such that $\forall x > y$ in $\mathcal{A}$, $x \succ y$ in $\mathcal{A}$ if and only if $f(x) = f(y) + 1$. Aigner ([?]) makes the following observation which we here prove:

Lemma 2.1 *$\mathcal{P}(n)$ is graded by $f(\mathcal{A}) = |E(\mathcal{A})|$.*

Proof: Consider $\mathcal{A}, \mathcal{B} \in \mathcal{P}(n)$ where $\mathcal{A} > \mathcal{B}$ in $\mathcal{P}(n)$.

Suppose that $\mathcal{A} \succ \mathcal{B}$ in $\mathcal{P}(n)$. Consider any injection of $\mathcal{B}$ into $\mathcal{A}$. There must be $x, y \in \mathcal{A} \ni x \succ y$ in $\mathcal{A}$, and $x \not\succ y$ in the embedding of $\mathcal{B}$ in $\mathcal{A}$ else $\mathcal{A} \simeq \mathcal{B}$. Thus, $\mathcal{A} > \mathcal{A} \setminus (x, y) \geq \mathcal{B}$. Hence, $|E(\mathcal{A})| = |E(\mathcal{A} \setminus (x, y))| + 1 \geq |E(\mathcal{B})|$. If there was another cover $u \succ v$ in $\mathcal{A} \setminus (x, y)$ not in the embedding of $\mathcal{B}$ in $\mathcal{A}$ then there would exist a poset $\mathcal{C} = (\mathcal{A} \setminus (x, y)) \setminus (u, v) \in \mathcal{P}(n)$ such that $\mathcal{A} > \mathcal{C} > \mathcal{B}$ in $\mathcal{P}(n)$, which implies that $\mathcal{A} \not\succ \mathcal{B}$, a contradiction. Hence, there can only be one relation in $\mathcal{A}$ not in any injection of $\mathcal{B}$ in $\mathcal{A}$ and thus, $|E(\mathcal{A})| = |E(\mathcal{B})| + 1$.

Conversely, suppose that $|E(\mathcal{A})| = |E(\mathcal{B})| + 1$. Suppose that there exists a $\mathcal{C}$ such that $\mathcal{A} > \mathcal{C} > \mathcal{B}$ in $\mathcal{P}(n)$. Then, $|E(\mathcal{A})| > |E(\mathcal{C})| > |E(\mathcal{B})|$. Which implies that $|E(\mathcal{A})| > |E(\mathcal{B})| + 1$, a contradiction. Thus, $\mathcal{A} \succ \mathcal{B}$ in $\mathcal{P}(n)$. ∎

Thus, $\mathcal{P}(n)$ is graded into *levels* by the number of relations function and $length(\mathcal{P}(n)) = \binom{n}{2}$. We refer to the set of posets with k relations in $\mathcal{P}(n)$ as the kth *level*.

These observations are critical to the following algorithms used in our computer enumeration of posets.

3 A Breadth-First Approach

The obvious method of counting posets, is to begin with $\mathcal{U}_n$ and generate all possible sets of relations. However, there are $2^{\binom{n}{2}}$ possible sets of relations, assuming that we only allow those compatible with some underlying total order. Most of these will violate the transitivity constraint, and so are not partial orders. In addition, many of the results will be isomorphic, and we wish to count only up to isomorphism. This method can be thought of as a bottom up approach.

Our approach is to construct $\mathcal{P}(n)$, using a top down approach. We start with $\mathcal{R}_n$. Note that $|E(\mathcal{R}_n)| = \binom{n}{2}$. It is easy to show that deleting a cover from a poset produces a new poset on the original set of elements. We cannot delete a non-covering relation, since this would violate transitivity. Thus, we may construct all of the posets in $\mathcal{P}(n)$ with $\binom{n}{2} - 1$ relations by deleting in turn each of the $n-1$ covers of $\mathcal{R}_n$.

Similarly, we may construct all of the posets with $\binom{n}{2} - 2$ relations by deleting in turn each of the covers from each of the posets with $\binom{n}{2} - 1$ relations. Note that some of these posets will be isomorphic; that is, some posets will be generated more than once.

In general, to generate the set of all posets with k relations, $\binom{n}{2} > k \geq 0$, given all the posets with $k+1$ relations, we delete in turn each cover from each of the given posets. ¿From lemma 2.1 it follows that each poset with k relations and n elements will eventually be generated, albeit each may be generated several times.

To count the number of non-isomorphic posets, we must eliminate isomorphic copies. This need only be done on a level by level basis, since isomorphism can only hold between posets with equal numbers of relations. However, as n increases, the number of potential isomorphism tests becomes quite large. The high cost of isomorphism testing can be mitigated somewhat by using a signature scheme based on hashing. As each poset is generated (by cover deletion from a poset on the previous level) it is hashed using a hash key based on such factors as the length of the poset and degree sequences. Posets in which these keys differ cannot be isomorphic, and will, in general, hash to distinct locations.

The top down approach has several advantages. Only those structures which are posets are constructed. Furthermore, if on constructing the posets of level k, we retain only one representative of all isomorphic posets, then for levels less than k, we never produce the (isomorphic) descendents of those not retained. This greatly reduces the number of redundant posets produced.

This level by level construction of $\mathcal{P}(n)$ is a breadth-first construction. We must at any time be prepared to retain at least one level of $\mathcal{P}(n)$ in memory for the program to be efficient, and to allow easy hashing. Since the number of posets in the widest level grows rapidly with n, this placed an upper limit on the size of posets we are able to count using this approach.

4 A Depth-First Approach

To construct $\mathcal{P}(n)$ in a depth-first manner, new insights are needed. The depth-first algorithm constructs an ordered rooted tree spanning $\mathcal{P}(n)$, with root node $\mathcal{R}_n$, in which the children of a node are the (non-isomorphic) posets created from it by deleting its covers ordered from left to right by order of creation.

More formally, the search tree is an ordered rooted tree, and if $\mathcal{A}$ is to the left of $\mathcal{B}$, ($\mathcal{A} <_l \mathcal{B}$) then all descendents of $\mathcal{A}$ are to the left of all descendents of $\mathcal{B}$. Thus, the left to right order provides a partial order on the set of posets, where if $\mathcal{A}$ is a descendent of $\mathcal{B}$, $\mathcal{A}$ and $\mathcal{B}$ are incomparable.

Let $\mathcal{R}$ be a poset at some node of the search tree. We use $\mathcal{R}$ to designate either the node or the poset, with the context supplying the proper interpretation. Let X be the set of covers in $\mathcal{R}$. Then, the algorithm specifies some ordering of the covers in X, say

$$x_1 \succ y_1\ ,\quad x_2 \succ y_2\ ,\quad \ldots\ ,\quad x_{|X|} \succ y_{|X|}$$

The set of posets $\{\mathcal{R} \setminus (x_i, y_i) | (x_i, y_i) \in X\}$ are the *potential children* of $\mathcal{R}$, with the subscript indicating the left to right ordering. A potential child is a child if there does not exist a node to the left of the potential child isomorphic to it. Thus, once the order of deletion is specified, the tree is unique.

The key idea in developing an efficient depth-first algorithm is to note that it is not necessary to know all nodes to the left of a potential node to determine if there is one isomorphic to it.

Lemma 4.1 $\exists \mathcal{C} <_l \mathcal{R} \setminus (x_i, y_i) \ni \mathcal{C} \simeq \mathcal{R} \setminus (x_i, y_i)$ *iff* $\exists \mathcal{A}, \mathcal{A} <_l \mathcal{B}$, *where* $\mathcal{A}$ *is a sibling of* $\mathcal{B}$ *and* $\mathcal{B}$ *is an ancestor of* $\mathcal{R} \setminus (x_i, y_i)$ *and* $\mathcal{A} \geq \mathcal{R} \setminus (x_i, y_i)$.

Proof: (Note that for any node $\mathcal{R}$, $\mathcal{R}$ is an ancestor of itself.)

If $\exists \mathcal{C} <_l \mathcal{R} \setminus (x_i, y_i) \ni \mathcal{C} \simeq_f \mathcal{R} \setminus (x_i, y_i)$, then $|E(\mathcal{C})| = |E(\mathcal{R} \setminus (x_i, y_i))|$. Thus, $\mathcal{C}$ and $\mathcal{R} \setminus (x_i, y_i)$ must have some common ancestor, say $\mathcal{D}$, distinct from both of them. Let $\mathcal{A}$ be the child of $\mathcal{D}$ which is an ancestor of $\mathcal{C}$, and let $\mathcal{B}$ be the child of $\mathcal{D}$ which is an ancestor of $\mathcal{R} \setminus (x_i, y_i)$. Now, $\mathcal{A} <_l \mathcal{B}$. Since $\mathcal{C}$ is derived by a sequence of deletions, then $\mathcal{C} \leq \mathcal{A}$. But then $\mathcal{R} \setminus (x_i, y_i) \leq \mathcal{A}$ under some embedding f.

On the other hand if $\mathcal{R} \setminus (x_i, y_i) \leq \mathcal{A}$, then if not equality, there exist relations in $\mathcal{A}$, (x, y) such that $f(x), f(y)$ is not a relation in $\mathcal{R} \setminus (x_i, y_i)$. If equality holds then, trivially, the lemma is true. At least one such relation must be a cover, since $f(\mathcal{R} \setminus (x_i, y_i))$ is a partial order. Deleting this cover produces a poset $\mathcal{A}'$, where the same argument applies. Eventually, we reach a poset isomorphic to $\mathcal{R} \setminus (x_i, y_i)$. ∎

This lemma allows us to do a depth-first search wherein we only keep track of the children of the ancestors of the current node. Each time we create a new poset, we check whether the current poset is contained in any of those children. The maximum number of children that can be required along any path is at most $\binom{n}{2}$, since that is the maximum number of relations. In fact, a more careful analysis yields an upper bound of about one-half of this. As a concrete example, this means that for $n = 10$ we only need at most 23 posets in memory at any time, (in addition to at most 45 ancestors) whereas, for the breadth-first algorithm we need two levels, each with potentially over 200,000 posets.

The disadvantage of this depth-first search is that testing poset containment is generally more difficult than testing isomorphism. Furthermore, we have no scheme analogous to hashing to eliminate the majority of containment tests. Fortunately, the number of children is usually much less than the upper bounds given above. Nevertheless, without further insight, many redundant tests must be made. In the next subsection we demonstrate how to eliminate most of these redundant tests.

4.1 A Deeper Analysis

Consider the poset $\mathcal{A}$ obtained by deleting some cover, (x, y), from some node $\mathcal{R}$. Then every poset that is contained in $\mathcal{A}$ is a potential descendent of $\mathcal{A}$. Now consider any other potential descendent $\mathcal{B}$ of $\mathcal{R}$, $\mathcal{A} <_l \mathcal{B}$, for which $\mathcal{A}$ is not an ancestor. If (x, y) is not a relation in $\mathcal{B}$, then $\mathcal{B}$ is isomorphic to a potential descendent of $\mathcal{A}$, by lemma 4.1. Thus, we need never delete (x, y) in any other descendent of $\mathcal{R}$ to the right of $\mathcal{A}$. We say that the cover (x, y) has been *blocked* after the generation of the subtree $\mathcal{A}$.

Similarly, if there is an automorphism (consistent with $\mathcal{R}_n$) which carries (x, y) to any other cover in $\mathcal{R}$, then that cover need not be deleted in any descendent of $\mathcal{R}$ to the right of $\mathcal{A}$, since such a deletion would lead to a poset isomorphic to a potential descendent of $\mathcal{A}$. Thus, all such automorphic covers can also be blocked. We can extend this to say that in any node, if a cover is carried to a blocked cover by some automorphism, then it too should be blocked. Note that a cover remains blocked for all subsequent descendents of the node in which the blocking occurred, even if in the descendent the automorphism which caused the blocking no longer holds because of deletions in the interim.

We represent the partial orders by lower triangular matrices M, in which:

- $M_{ij} = 0$ means that $x_i \parallel x_j$,
- $M_{ij} = A$ means that $x_i \parallel x_j$ and $[x_i, x_j]$ is an automorphism (see below),
- $M_{ij} = C$ means that $x_i \succ x_j$,
- $M_{ij} = X$ means that $x_i \succ x_j$ but the cover is blocked, and
- $M_{ij} = 1$ means that $x_i > x_j$ but the relation is not a cover.

See Figure 2. (We will explain the meaning of 'P' shortly. This example does not record automorphisms). Note that only the lower triangular matrix NOT including the diagonal is presented.

Initially we represent $\mathcal{R}_n$ by such a matrix with all subdiagonal elements 'C', and the remainder all '1'. When a cover is deleted, some of the non-covering relations may become covers in the new subposet. Note that, since the matrix is lower triangular, every poset has the linear extension $\mathcal{R}_n$.

To be certain that a cover is not blocked could require that it be tested against every other blocked cover to see if there is an automorphism. In our program we do not do full automorphism checking, but are content to consider only those automorphisms which switch pairs of elements; for example, $f(a) = b$ and $f(b) = a$. We denote such pairs by $[a, b]$. Thus, a cover (x, y) is automorphic to a cover (z, w) if there exist automorphic pairs $[x, z]$ and $[y, w]$, or one of the pairs if the others are equal.

We use an automorphism $[a, b]$ only if rows a and b are identical and columns a and b are identical (where, for this purpose, 'X'='C' and 'A'='0') and $a \parallel b$. Only the rows and columns in question need be checked, and so this is an $O(n)$ operation.

To record such automorphisms, we store an 'A' in the matrix at M_{ab}. When a new poset is created by deleting some cover (x, y), we can update the automorphism set by checking rows and columns x and y for occurrences of 'A', and if any are found they are changed to '0' since that automorphism can no longer apply. If '0' is found, then it is checked as a possible new automorphism. $[x, y]$ may also be an automorphism. Thus, updating the automorphism set takes $O(n^2)$ time.

For each new automorphism $[x, y]$ generated we check all blocked covers (x, y), (x, z), (z, x), (z, y) and (y, z) (for each z) to determine if they are automorphic to any other (possibly new) covers. All of this can be done in $O(n^2)$ time as well. The complete algorithm is outlined in Figure 1.

How effective is this in reducing the number of containment checks? This depends upon the order in which the covers are deleted from the posets. With no blocking, 1,982,737 posets were generated for $n = 9$. ¿From Table 1 we see that there are 183,231 distinct posets in this case, so we generated more than ten times as many as required, and each of these was eliminated by some containment test. By deleting the covers diagonally from top to bottom, left to right, and using automorphisms to block covers, this was reduced to 204,468. This means that we have generated at most 21,237 extras, or about 11.5%. For $n = 10$, this is reduced to about 8.5%, and for $n = 11$ it is 5.9%.

The order in which we do deletions does affect the efficiency of the automorphisms in reducing the posets generated. The diagonal order seems to be better than row by row or column by column deletion orders. The intuition is that deleting an element from the diagonal yields a high probability of an automorphism, and the sooner this occurs, the more effective will be the pruning of the tree we build.

The question occurs, if we generated all possible automorphisms, could we eliminate the containment test altogether? That is, would considering all automorphisms block all isomorphic posets? The answer to this is unfortunately no, as the example on 6 elements in Figure 2 demonstrates for the diagonal deletion order. (A different deletion order might fail elsewhere).

Recall that only the lower triangular matrix not including the diagonal is presented. Thus row a and column f are not shown. The example starts after 6 deletions have been completed, with the seventh poset generated by the algorithm. Each poset is a child of the poset above it, with the cover indicated that was deleted to produce it from its parent. The numbers indicate the ordinals of the posets in the order of generation by the program. The letter 'P' is used to indicate a cover that was deleted to produce the missing descendents that are not shown. Thus, 'P' is in fact a blocked cover. 'X' indicates a cover that is blocked because it is automorphic (or was in some ancestor) to a 'P'.

As can be easily seen, posets 57 and 69 are isomorphic. The isomorphism $f : 69 \to 57$ is $f(a) = c, f(b) = b, f(c) = a, f(d) = f, f(e) = e, f(f) = d$. However, there is no automorphism in any ancestor of 69 which could block the cover (f, a).

5 Results

For $n = 10$, we obtain 2,567,284 posets, while Möhring [?] found only 2,567,249 posets. To verify the extra posets, we recomputed the results for $n = 10$, this time storing compact representations of each poset generated in a database. We then ran a testing program which selected all the posets with a given number of relations and checked to see if any two were isomorphic (again using hashing to eliminate the obviously distinct posets). In addition, a closure test was performed on each poset to ensure that the objects represented were (transitive) partial orders. Since they are stored as a lower triangular matrix, they must be directed and acyclic. The generation and testing for $n = 10$ each required about 2 days on a Sun 3/50. No isomorphic posets were encountered.

Counting for $n = 11$ required 112 hours on a MIPS Computer Systems M/1000. The program keeps track of the number of posets for r relations and n elements. The complete table for $n \leq 11$ is shown in Table 1.

In Table 2 we list the number of posets with δ relations deleted from $\mathcal{R}_n$. This is essentially the bottom of Table 1, but extended to larger n (and inverted). Interesting recurrences for small δ seem to hold, when $n > \delta$. We let $T_{\delta n}$ be the number of posets on n elements with $\binom{n}{2} - \delta$ relations. Then,

$$\begin{aligned}
T_{0,n} &= T_{0,n-1} \\
T_{1,n} &= T_{1,n-1} + T_{0,n-1} \\
T_{2,n} &= T_{2,n-1} + T_{1,n-1} + 0T_{0,n-1} \\
T_{3,n} &= T_{3,n-1} + T_{2,n-1} + 0T_{1,n-1} + 2T_{0,n-1} \\
T_{4,n} &= T_{4,n-1} + T_{3,n-1} + 0T_{2,n-1} + 2T_{1,n-1} - 1T_{0,n-1} \\
T_{5,n} &= T_{5,n-1} + T_{4,n-1} + 0T_{3,n-1} + 2T_{2,n-1} - 1T_{1,n-1} + 5T_{0,n-1}
\end{aligned}$$

We note that $T_{0n} = 1$, $T_{1n} = n-1$ and $T_{2n} = \binom{n-1}{2}$. In general, the recurrence is

$$T_{\delta,n} = \sum_{i=0}^{\delta} C_{\delta-i} T_{i,n-1} \; for \; n > \delta$$

The observed coefficient sequence is

$$C = 1, 1, 0, 2, -1, 5, -5, 19, -31, 92, -193, 525, -1252, 3321, -8427, \ldots$$

We do not know how to predict this sequence, or to prove in general that the recurrence holds, although we show below that it is correct for the first few terms.

Another similar result is the following sequence of functions:

$$\begin{aligned}
T_{0,n} &= T_{0,n-1} \\
T_{1,n} &= T_{1,n-1} + T_{0,n-2} \\
T_{2,n} &= T_{2,n-1} + T_{1,n-2} + T_{0,n-3} \\
T_{3,n} &= T_{3,n-1} + T_{2,n-2} + T_{1,n-3} + 3T_{0,n-4} \\
T_{4,n} &= T_{4,n-1} + T_{3,n-2} + T_{2,n-3} + 3T_{1,n-4} + 8T_{0,n-5} \\
T_{5,n} &= T_{5,n-1} + T_{4,n-2} + T_{3,n-3} + 3T_{2,n-4} + 8T_{1,n-5} + 21T_{0,n-6}
\end{aligned}$$

Again, the pattern of coefficients remains the same (except for the new term added) for each successive level, and appears valid for $n > \delta$. The relationship between the two formulations should be clear. In this case, the coefficients appear to be all positive and the sequence that can be derived from the table in Figure 2 is 1, 1, 1, 3, 8, 21, 63, 195, 612, 1971, 6458, 21426, 71905, 243640 ... Since only when we delete $n-1$ or more edges can we obtain a poset consisting of more than one component, we suspect that the fact that the rules do not apply for $n \leq \delta$ may be related to the presence of components.

If we set up a matrix B by working backwards from the rightmost column of Table 2, we can compute new values for the table when $n \leq \delta$. Taking the difference $T - B$ of these two matrices, leaves us with the lower triangle matrix shown in figure 3. Note that the values along the upper diagonal are the Fibonacci sequence. If we let $t_i^{(k)}$ be the element in the ith column in the kth non-zero diagonal from the top, then the following patterns can be discerned.

- $t_i^{(1)} = t_{i-1}^{(1)} + t_{i-2}^{(1)}$, $i \geq 3$ (the Fibonacci sequence)
- $t_i^{(2)} = t_{i-1}^{(2)} + t_{i-2}^{(2)} + t_i^{(1)}$, $i \geq 3$
- $t_i^{(3)} = t_{i-1}^{(3)} + t_{i-2}^{(3)} + t_i^{(2)} - t_{i-3}^{(1)}$, $i \geq 4$

We now verify the first three terms of the coefficient sequence C. Trivially, $T_{0j} = 1$, since there is only one total order. Similarly, it is easy to verify that $T_{1j} = j-1$, since there are $j-1$ covers in the total order, and deleting each one leads to a unique poset.

We also verify that $T_{2j} = \binom{j-1}{2}$. There are $n-1$ covers along the subdiagonal, and we get to delete 2 of them. However, after deleting one cover from the subdiagonal, say $(x, x-1)$, up to 2 new covers are created, in $(x, x-2)$ and $(x+1, x-1)$. (When x is the second or last row, only one new cover is formed). Thus, one might assume the above count is incomplete. But it is easily seen that $[x, x-1]$ is an automorphism of the matrix after the deletion, and thus deleting $(x, x-2)$ is equivalent to deleting the subdiagonal cover $(x-1, x-2)$ and deleting $(x+1, x-1)$ is equivalent to deleting the subdiagonal cover $(x+1, x)$ (when such exist). Since this is true for any x, we need only choose 2 covers from the $n-1$ on the diagonal. Each such pair yields a distinct poset.

Our program is easily modified to compute the number of lower triangular matrices which represent posets. The number of these is the number of natural partial orders, one instance of a class of lattices that have received some attention in the literature [?,?,?]. This problem does not involve any isomorphism or containment testing. It is only necessary to be certain that each subset of deleted edges is counted only once, which is easily done by blocking a cover but not any of its isomorphisms.

We give the results for $n \leq 9$ in Table 4. These results agree with those of Avann [?] for $n \leq 5$. In Table 5 we show the number of such matrices which represent posets with δ covers deleted from the total order. As in the number of posets, for $n \geq \delta$ we can generate the table $M_{\delta n}$ using the following recurrences.

$$\begin{aligned}
M_{0,n} &= M_{0,n-1} \\
M_{1,n} &= M_{1,n-1} + M_{0,n-1} \\
M_{2,n} &= M_{2,n-1} + M_{1,n-1} + 2M_{0,n-1} \\
M_{3,n} &= M_{3,n-1} + M_{2,n-1} + 2M_{1,n-1} + 3M_{0,n-1} \\
M_{4,n} &= M_{4,n-1} + M_{3,n-1} + 2M_{2,n-1} + 3M_{1,n-1} + 6M_{0,n-1} \\
M_{5,n} &= M_{5,n-1} + M_{4,n-1} + 2M_{3,n-1} + 3M_{2,n-1} + 6M_{1,n-1} + 8M_{0,n-1}
\end{aligned}$$

We note that $M_{0,n} = 1$, $M_{1,n} = n-1$, and $M_{2,n} = \binom{n+1}{2} - 3$. Avann [?] found an equivalent recurrence for the first few rows, but could not generalize his observed pattern. The general recurrence is

$$M_{\delta n} = \sum_{i=0}^{\delta} K_{\delta - i} T_{i,n-1}$$

The sequence we find is

$$K = 1, 1, 2, 3, 6, 8, 17, 24, 39, 78, 103, 184, 313, \ldots$$

In this case the sequence appears to have all positive terms. We can also obtain the sequence for the diagonal recurrence as before which is 1,1,3,10,39,159,685,3042,13860,64393,303949,1453428, ... Again, we can do a backwards computation using this matrix and the values in the last column and first row of Table 5 and take the difference to obtain Table 6.

In this case, the first non-zero diagonal appears to be powers of two. If we let $m_i^{(k)}$ be the value in the ith column of the kth non-zero diagonal, then we find the following patterns.

- $m_i^{(1)} = 2^i$, $i \geq 1$
- $m_i^{(2)} = 2^{i-3}(7i + 12)$, $i \geq 2$
- $m_i^{(3)} = 2^{i-7}(7^2 i^2 + 7^3 i - 124)$, $i \geq 4$

Again we offer no proof or explanation for these sequences, although the first few rows of $M_{\delta n}$ are easy to verify.

¿From a different program we obtained the number of posets on r relations, where $n \geq 2r$. We computed the number of connected posets on r relations as well as the total number of posets and derived the number of posets with 2 or more components.

Number of Relations	0	1	2	3	4	5	6	7	8	9
Number Connected	1	1	2	4	9	20	54	134	383	1092
Number with Components	0	0	1	3	10	27	79	221	640	1855
Total	1	1	3	7	19	47	133	355	1023	2947

6 Conclusion

In this paper we present a new algorithm for counting partial orders. We have verified results of Möhring [?] for $n \leq 9$ but claim an additional 35 posets for $n = 10$ and give results for $n = 11$. We also extend the work of Avann [?] from $n = 5$ to $n = 9$ in counting the number of natural partial orders. We have also discovered some interesting if unexplained recurrences that appear to hold for the number of partial orders and for the number of natural partial orders. The very strong patterns exhibited could be indicative of some means of enumerating partial orders, or at least those with $\binom{n}{2} - \delta$ relations, for $\delta < n$.

We would be most interested in hearing of a predictive function for the coefficients of these recurrences.

abbrv

7 Appendix

An alternative to counting is to find an asymptotic formula for the number of partial orders. Several researchers have contributed to this effort. Combining results from [?,?] and [?] (ex. 3, pg. 154) we see that

$$|\mathcal{P}(n)| \sim \frac{A}{n}\left(\frac{e}{n}\right)^n 2^{n^2/4+3n/2}\,, \quad \text{where } A \approx 0.80587793 (n \text{ even})$$

The following table compares these results with our counts. The asymptotic formula does not appear to be a good predictor for small n.

```
{ Branches is a global stack of posets, which are children of ancestors of the current poset.}
Main;
  Initialize Poset to R_n;
  For each cover C in Poset do
    begin
      Search(Poset, C, SuccessFlag, ReducedPoset);
      Block C in Poset;
      Push ReducedPoset onto Branches; { Always successful }
    end
  Print results;
end
Procedure Search(Poset, Cover, Var SuccessFlag, Var ReducedPoset);
  { If the poset formed by deleting Cover from Poset is new, }
  { then it is copied to ReducedPoset and SuccessFlag is set true }
  Delete Cover from Poset;
  Update the Set of Covers in Poset;
  If Poset is not a subposet of any poset in Branches then
    begin
      SuccessFlag := true;
      ReducedPoset := Poset;
      Increment the number of Posets;
      Update the set of automorphisms in Poset;
      Update the set of blocked covers in Poset;
      For each unblocked cover C in Poset do
        begin
          Search(Poset, C, Flag, Red_Pos);
          Block C in Poset;
          Block all covers automorphic to C in Poset;
          If ( Flag ) then
            Push Red_Pos onto Branches;
        end;
    end;
  Pop all the posets added to Branches during this call;
end
```

Figure 1: The Poset Counting Algorithm

7

	a	b	c	d	e
b	0				
c	0	0			
d	C	C	0		
e	C	C	C	0	
f	1	1	C	C	0

8 (d,b)

	a	b	c	d	e
b	0				
c	0	0			
d	C	0	0		
e	C	C	P	0	
f	1	C	C	C	0

65 (e,c)

	a	b	c	d	e
b	0				
c	0	0			
d	X	X	0		
e	C	C	0	0	
f	1	1	C	C	0

54 (f,d)

	a	b	c	d	e
b	0				
c	0	0			
d	P	0	0		
e	C	X	X	0	
f	C	X	X	0	0

66 (f,d)

	a	b	c	d	e
b	0				
c	0	0			
d	X	X	0		
e	X	X	0	0	
f	C	C	P	0	0

56 (e,a)

	a	b	c	d	e
b	0				
c	0	0			
d	X	0	0		
e	0	X	X	0	
f	C	X	X	0	0

68 (f,b)

	a	b	c	d	e
b	0				
c	0	0			
d	X	X	0		
e	X	X	0	0	
f	C	0	X	0	0

57 (f,a)

	a	b	c	d	e
b	0				
c	0	0			
d	X	0	0		
e	0	X	X	0	
f	0	X	X	0	0

69 (f,a)

	a	b	c	d	e
b	0				
c	0	0			
d	X	X	0		
e	X	X	0	0	
f	0	0	X	0	0

Figure 2: Counter Example to the Automorphism Conjecture

$r \setminus n$	1	2	3	4	5	6	7	8	9	10	11		10	11
0	1	1	1	1	1	1	1	1	1	1	1	37	3240	599653
1		1	1	1	1	1	1	1	1	1	1	38	1836	430451
2			2	3	3	3	3	3	3	3	3	39	986	300981
3			1	4	6	7	7	7	7	7	7	40	506	204974
4				3	10	16	18	19	19	19	19	41	237	135976
5				3	10	25	38	44	46	47	47	42	99	87786
6				1	12	36	74	107	124	130	132	43	36	55127
7					9	43	113	208	287	329	346	44	9	33614
8					6	46	167	381	636	841	950	45	1	19897
9					4	44	209	619	1257	1946	2468	46		11385
10					1	35	243	915	2311	4251	6171	47		6306
11						28	249	1219	3830	8526	14411	48		3351
12						17	239	1506	5891	15891	31724	49		1694
13						10	204	1705	8294	27259	64772	50		811
14						5	168	1792	10921	43572	123620	51		353
15						1	123	1767	13363	64851	219868	52		137
16							83	1621	15419	90614	366672	53		45
17							54	1402	16687	119179	574347	54		10
18							29	1136	17119	148255	849968	55		1
19							15	874	16578	174838	1190889			
20							6	629	15309	196135	1587016			
21							1	434	13421	209729	2015412			
22								274	11253	214283	2446957			
23								166	8999	209692	2844542			
24								94	6897	196824	3174558			
25								46	5054	177576	3405232			
26								21	3551	154148	3518608			
27								7	2386	128998	3505930			
28								1	1528	104101	3374784			
29									939	81200	3141073			
30									541	61145	2831400			
31									300	44566	2473385			
32									153	31401	2096755			
33									69	21414	1725908			
34									28	14096	1380922			
35									8	8974	1074413			
36									1	5492	813564			
	1	2	5	16	63	318	2045	16999	183231				2567284	46749427

Table 1: The Number of Posets on n Elements and r Relations

$\delta \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	0	0	1	3	6	10	15	21	28	36	45	55	66	78	91
3	0	0	1	4	9	17	29	46	69	99	137	184	241	309	389
4	0	0	0	3	12	28	54	94	153	237	353	509	714	978	1312
5	0	0	0	1	10	35	83	166	300	506	811	1249	1862	2701	3827
6	0	0	0	1	10	44	123	274	541	986	1694	2779	4391	6724	10025
7	0	0	0	0	6	46	168	434	939	1836	3351	5808	9660	15527	24242
8	0	0	0	0	3	43	204	629	1528	3240	6306	11545	20165	33907	55229
9	0	0	0	0	1	36	239	874	2386	5492	11385	21985	40260	70684	119854
10	0	0	0	0	1	25	249	1136	3551	8974	19897	40515	77647	142020	250105
11	0	0	0	0	0	16	243	1402	5054	14096	33614	72382	145258	276596	505213
12	0	0	0	0	0	7	209	1621	6897	21414	55127	125818	264558	524242	992262
13	0	0	0	0	0	3	167	1767	8999	31401	87786	213115	470396	970342	1902199
14	0	0	0	0	0	1	113	1792	11253	44566	135976	352196	817624	1757459	3568480

Table 2: The Number of Posets on n Elements with δ Relations Deleted from $\mathcal{R}_n$

$\delta \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	-3	1	1	0	0	0	0	0	0	0	0	0	0	0	0
5	16	8	5	2	0	0	0	0	0	0	0	0	0	0	0
6	-29	-2	8	8	3	0	0	0	0	0	0	0	0	0	0
7	99	42	27	24	16	5	0	0	0	0	0	0	0	0	0
8	-237	-57	17	46	47	29	8	0	0	0	0	0	0	0	0
9	686	253	123	112	121	99	53	13	0	0	0	0	0	0	0
10	-1828	-570	-76	134	242	263	197	95	21	0	0	0	0	0	0
11	5114	1813	693	458	535	631	579	388	169	34	0	0	0	0	0
12	-14135	-4847	-1311	73	794	1281	1449	1222	749	298	55	0	0	0	0
13	39512	14132	4941	2245	2042	2718	3347	3310	2535	1427	522	89	0	0	0
14	-110775	-39759	-12847	-2870	1517	4502	6892	7988	7285	5153	2685	909	144	0	0

Table 3: The Differences of the Count of Posets and the Backwards Computation

$r \setminus n$	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1
1		1	3	6	10	15	21	28	36
2			2	11	35	85	175	322	546
3			1	11	65	260	805	2086	4746
4				7	81	526	2436	8911	27363
5				3	70	766	5348	27636	114618
6				1	51	869	9041	66007	370031
7					27	809	12407	126928	959688
8					12	633	14291	203384	2064638
9					4	423	14234	279175	3779795
10					1	243	12452	335558	6014986
11						120	9723	359588	8469363
12						50	6753	348051	10713624
13						18	4234	306795	12327354
14						5	2391	247714	13027980
15						1	1221	184106	12736837
16							555	126464	11581866
17							227	80469	9835487
18							81	47498	7827305
19							25	26019	5853169
20							6	13222	4123506
21							1	6215	2741008
22								2688	1721422
23								1063	1021917
24								382	573767
25								121	304502
26								33	152652
27								7	72113
28								1	32030
29									13308
30									5149
31									1836
32									596
33									171
34									42
35									8
36									1
	1	2	7	40	357	4824	96428	2800472	116473461

Table 4: The Number of Lower Triangular Matrices Representing Posets

$\delta \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	2	3	4	5	6	7	8	9	10	11	12
2	0	0	3	7	12	18	25	33	42	52	63	75	88
3	0	0	1	11	27	50	81	121	171	232	305	391	491
4	0	0	0	11	51	120	227	382	596	881	1250	1717	2297
5	0	0	0	6	70	243	555	1063	1836	2956	4519	6636	9434
6	0	0	0	1	81	423	1221	2688	5149	9023	14838	23247	35045
7	0	0	0	0	65	633	2391	6215	13308	25439	45002	75156	119983
8	0	0	0	0	35	809	4234	13222	32030	67033	127667	227192	383758
9	0	0	0	0	10	869	6753	26019	72113	166295	341680	648144	1157895
10	0	0	0	0	1	766	9723	47498	152652	390420	868262	1757372	3320397
11	0	0	0	0	0	526	12452	80469	304502	870691	2104205	4552390	9100974
12	0	0	0	0	0	260	14234	126464	573767	1849790	4880559	11312182	23950134

Table 5: The Number of Matrices Representing Posets on n Elements with δ Relations Deleted from $\mathcal{R}_n$

$\delta \setminus n$	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	0	0
3	5	4	0	0	0	0	0	0	0	0	0	0	0
4	10	13	8	0	0	0	0	0	0	0	0	0	0
5	8	26	33	16	0	0	0	0	0	0	0	0	0
6	7	45	84	80	32	0	0	0	0	0	0	0	0
7	-26	49	168	254	188	64	0	0	0	0	0	0	0
8	-97	10	274	611	704	432	128	0	0	0	0	0	0
9	-145	-90	364	1243	2031	1849	976	256	0	0	0	0	0
10	-408	-454	237	2157	4878	6119	4678	2176	512	0	0	0	0
11	-479	-1025	-352	3205	10275	17074	17320	11512	4800	1024	0	0	0

Table 6: The Differences of the Number of Matrices and the Backwards Matrix Computation