Counting Linear Extensions

Graham Brightwell*

London School of Economics and Political Science

Houghton Street, London, U.K.

Peter Winkler

Bellcore

445 South St., Morristown, New Jersey

* Research carried out while this author was visiting Bellcore under the auspices of DIMACS.

Abstract.

We survey the problem of counting the number of linear extensions of a partially ordered set. We show that this problem is #P-complete, settling a long-standing open question. This result is contrasted with recent work giving randomized polynomial-time algorithms for *estimating* the number of linear extensions.

One consequence of our main result is that computing the volume of a rational polyhedron is strongly #P-hard. We also show that the closely related problems of determining the average height of an element x of a given poset, and of determining the probability that x lies below y in a random linear extension, are #P-complete.

1. Introduction.

The problem of determining the number of linear extensions of a partially ordered set is fundamental in the theory of ordered sets, and is of interest in computer science by virtue of its connections with sorting. For instance, at each stage of any comparison-based sorting algorithm, current information can be expressed as a partial ordering of the data set, any linear extension of which is a possible "solution". If it were easy to compute the number of linear extensions of a poset, one could determine in a sequential sort which is the optimum pair of elements to compare next. (Kahn and Saks [15] have shown that there is always a pair whose comparison splits the set of linear extensions in no more lopsided a fashion than $3/11 : 8/11$, but their proof gives no way of finding such a pair short of computing numbers of linear extensions.)

Another application arises in the social sciences, when a ranking of alternatives (e.g., products, job candidates, athletes in a competition) must be determined from a partial ordering (see, e.g., [11]). One natural such ranking is given by "average height", in which each element is assigned the mean of its ranks in all linear extensions. This approach has the advantage that certain desirable correlations are achieved; for example, additional information to the effect that element x is below element y can only drive the average height of x down, and y up (see [30]). Here again, however, numbers of linear extensions must apparently be computed; and since such numbers may be exponential in the number of elements, it is not clear that efficient methods can be found.

The problem of counting linear extensions of a partial order can also be regarded as a special case of the problem of calculating the volume of a polyhedron in n-dimensional space, as we shall see later.

Unfortunately from a theoretical point of view, determining the difficulty of counting linear extensions has itself been frustratingly difficult. In [2–4,12,25], polynomial algorithms are given for counting linear extensions under various special circumstances: for example, in cases where the poset is a tree, is series-parallel, or has bounded width. The case of bounded *height*, and therefore the general case as well, remained unsolved.

Recently, however, randomized polynomial time algorithms have been given which *approximate* the number of linear extensions to within an abritrary tolerance. In 1989 Dyer, Frieze and Kannan [9], by applying their work on approximating the volumes of convex bodies to the order polytope of a poset, became the first to obtain such an algorithm. A much more efficient algorithm for estimating the number of linear extensions is now available, however, thanks to the bounds on conductance of the linear extension graph achieved by Karzanov and Khachiyan [16]. We shall discuss these results in more detail in Section 3.

On the other hand, the problem of determining exactly the number of linear extensions has long been suspected of being intractable. The problem is clearly in the class #P introduced by Valiant [28] in the 'seventies, since it is easy to check whether a linear ordering is consistent with the given poset. Since then Linear Extension Count has been widely suspected of being #P-complete, and thus probably very difficult (especially in view of Toda's result [27], which implies that one call to a #P oracle suffices to solve any problem in the polynomial hierarchy in deterministic polynomial time). As far as we know the first to make the conjecture in print was Linial [19], who called it "a most intriguing problem in this field." Lovász [20, p. 61] has also mentioned the problem, and apparently many others have considered it. It has, however, resisted analysis, despite the substantial number of related counting problems which have by now been shown to be #P-complete. These problems include counting antichains in a partial order [23], counting acyclic orientations of a graph [19], computing the number of linear extensions of certain special types [18,26], and computing the volume of a general convex body in Euclidean space [7]. The last of these is now known to be #P-complete "in the strong sense" (see Khachiyan [17]) as a consequence of our result.

Our method is direct, showing that with the help of an oracle which counts linear extensions, a Turing machine can count the number of satisfying assignments to an instance of 3-SAT in polynomial time. This contrasts with other #P-completeness results, as in [19,23], which have utilized the machinery developed in Valiant [28,29]. We do however follow Valiant in approaching the problem via enumeration modulo many different primes. Our technique has now been applied by Feigenbaum and Kahn [10] to show that a problem called "POMSET language size" is complete for the class SPAN-P.

Much of this paper has previously been published in the form of an extended abstract [5].

2. Preliminaries.

A partially ordered set (or poset) is a set P equipped with an irreflexive transitive relation $<$. An antichain in P is a set of elements (vertices) of P such that no pair is related by $<$.

A *linear extension* of a partially ordered set P on n vertices is a linear ordering $\prec$ of the vertex set such that $x \prec y$ whenever $x < y$ in P. Equivalently, a linear extension of P is a bijection λ from the set of vertices of P to $\{1, \ldots, n\}$ such that $\lambda(x) < \lambda(y)$ whenever $x < y$ in P. We shall be making implicit use of both forms of the definition. For a poset P, let $\Lambda(P)$ denote the set of linear extensions of P, and set $N(P) = |\Lambda(P)|$, the number of linear extensions of P.

For x and y incomparable elements of a poset P, $\Pr(x \prec y \mid P)$ (or, briefly, $\Pr(x \prec y)$) denotes the probability that x precedes y in a randomly chosen linear extension of P, where all linear extensions are equally likely. Thus $\Pr(x \prec y \mid P)$ can be written as $N(P \cup (x, y))/N(P)$, where $P \cup (x, y)$ is the poset obtained from P by adding the relation $x < y$ and taking the transitive closure.

We shall mostly be concerned with the following enumeration problem.

Linear Extension Count

Input. A partially ordered set P.

Output. The number $N(P)$ of linear extensions of P.

The complexity class #P consists of all counting problems whose solutions are the number of accepting states of some nondeterministic polynomial time Turing machine. In this paper, we shall make use of the basic fact, proved in [28], that the following problem is #P-complete.

3-SAT Count

Input. A propositional formula I in 3-conjunctive normal form.

Output. The number $s(I)$ of satisfying assignments for I.

The main result in Valiant [28] is that computing the permanent of a matrix (equivalently, counting the number of complete matchings in a bipartite graph) is #P-complete. This remains the outstanding example of a case where a decision problem is in P, but the corresponding enumeration problem is #P-complete.

Our main result is another example of this phenomenon, even more extreme since the decision problem is trivial: every poset has a linear extension.

Theorem 1. *Linear Extension Count is #P-complete.*

The implication of Theorem 1 is that there is very unlikely to be a fast algorithm to count the exact number of linear extensions of a general partial order. In these circumstances, a reasonable alternative is to be able to approximate the number of linear extensions. This is indeed possible, and we wish to contrast Theorem 1 with the following result, due originally to Dyer, Frieze and Kannan [9].

Theorem 2. *There exists a randomized algorithm $\mathcal{A}$ with the following properties. The input consists of an n-element partial order P, and positive rational numbers ϵ, β. The output is a number L with the property that:*

$$\Pr\left(\left|\frac{L}{N(P)} - 1\right| < \epsilon\right) > 1 - \beta.$$

The algorithm runs in time polynomial in n, $1/\epsilon$ and $\log(1/\beta)$.

Such an algorithm is called a *fully polynomial randomized approximation scheme* or *fpras* for the problem Linear Extension Count. The interpretation is that algorithm $\mathcal{A}$ finds, with arbitrarily high probability, an approximation L to the number of the linear extensions which is within a multiplicative factor $(1+\epsilon)$ of the correct number.

We give more details concerning randomized algorithms for Linear Extension Count in Section 3. In Section 4, we give the proof of Theorem 1, and in the final section we discuss the consequences of Theorem 1 for some problems closely related to Linear Extension Count.

Later, we shall have need of the following technical lemma concerning the distribution of primes.

Lemma. *For any $n \geq 4$, the product of the set of primes strictly between n and n^2 is at least $n!\,2^n$.*

Proof. We use some facts from Hardy and Wright [13, Chapter 22] concerning the functions $\vartheta(n) = \log \prod_{p \leq n} p$, where p runs over all primes less than n, and $\psi(n) = \sum_{i=1}^{\log n/\log 2} \vartheta(n^{1/i})$. From [13] we find that $\vartheta(n) < 2n \log 2$ for $n \geq 1$, and that $\psi(n) \geq \frac{1}{4} n \log 2$ for $n \geq 2$.

We are interested in the quantity $V = \vartheta(n^2) - \vartheta(n)$. From the above facts, we have:

$$\begin{aligned} V &\geq \psi(n^2) - \sum_{i=2}^{2\log n/\log 2} \vartheta(n^{2/i}) - \vartheta(n) \\ &\geq \frac{1}{4} n^2 \log 2 - \frac{2 \log n}{\log 2} \cdot 2n \log 2 - 2n \log 2 \\ &\geq n \log n \geq \log(n!\,2^n), \end{aligned}$$

at least provided $n \geq 150$. The inequality for $4 \leq n < 150$ is easily verified by direct calculation. $\square$

It is evident that this lemma is not tight: it is possible to replace the n^2 upper limit by $Kn \log n$, for some suitably large K.

3. Approximating the number of linear extensions.

This section constitutes a survey of recent progress on the subject of approximation algorithms for the number of linear extensions. Our principal aim here is to bring this work to the atttention of researchers in the theory of ordered sets. We are not claiming any great originality for the results in this section, although not all the material appears in detail elsewhere. In particular, we give details of a fpras having a running time estimate with dominant term n^9, which, although unlikely to be anywhere near optimal, is a slight improvement on previous estimates for this problem.

We begin by noting the connection between counting linear extensions and computing the volume of a convex body. Given a partial order P on an n-element set $\{a_1, \dots a_n\}$, we define the *order-polytope* $Q(P)$ as

$$\{\underline{x} \in [0,1]^n \, : \, x_i < x_j \text{ whenever } a_i < a_j \text{ in } P\}.$$

What is the volume (n-dimensional Lebesgue measure) of $Q(P)$? Apart from a set of measure 0 where some pair of coordinates is equal, $[0,1]^n$ can be partitioned into sets

$$Q_\sigma = \{\underline{x} \in [0,1]^n \, : \, x_{\sigma(1)} < x_{\sigma(2)} < \dots < x_{\sigma(n)}\},$$

where σ is a permutation of $\{1, 2, \dots, n\}$. By symmetry, each of the Q_σ has the same volume $1/n!$. Now $Q(P)$ is made up (apart from a set of measure 0) of those Q_σ where $a_{\sigma(1)} \prec a_{\sigma(2)} \prec \dots \prec a_{\sigma(n)}$ is a linear extension of P. Thus the volume of $Q(P)$ is exactly $N(P)/n!$.

Therefore negative results about Linear Extension Count imply negative results about volume calculation, whereas positive results about the computation of volume give positive results about counting linear extensions.

In an important 1989 paper, Dyer, Frieze and Kannan [9] gave a fpras for approximating the volume of an n-dimensional convex body. (The model of computation is that one is given initially a small ball A and a large ball B such that the body K satisfies $A \subseteq K \subseteq B$, and the algorithm may consult a membership oracle for K: given a point $a \in \mathbf{R}^n$, the oracle reveals whether $a \in K$.)

Since the order polytope of an n-element partial order is a convex n-dimensional body, this immediately gives a fpras for Linear Extension Count, as noted in [9]. We should remark that, although the approximation scheme runs in time polynomial in n, the degree of the polynomial given in [9] is rather large. Karzanov and Khachiyan [16] and Lovász and Simonovits [21] have reduced this degree somewhat and, using a slightly more indirect approach, Applegate and Kannan [1] have developed a scheme where the dominant term in the run-time is only n^{10}. Matthews [22] has analyzed the special case of finding a random point in an order polytope. For a survey article concentrating on the problem of volume computation, the reader is referred to Dyer and Frieze [8]. For the remainder of this section, we shall concentrate on a scheme based on work of Karzanov and Khachiyan [16], which is tailored to Linear Extension Count. Perhaps it should be emphasized that none of the schemes mentioned above gives an approximation algorithm which is truly practical.

A feature common to several recent approximation schemes is that the problem of approximating the cardinality of a set is reduced to that of generating a member of that set at random according to an approximately uniform distribution. The approximate uniform generation is then accomplished by setting up a rapidly mixing Markov chain whose states are the members of the set. Besides those papers we have already mentioned, such a technique is used in Broder [6], and Jerrum and Sinclair [14] to produce a fpras for finding the number of complete matchings in a dense bipartite graph.

The contribution of Karzanov and Khachiyan [16] was to prove the rapid mixing property for a very natural Markov chain on the set of linear extensions of a partial order. Thus they were able to get a fast algorithm to approximate $\Pr(x \prec y \mid P)$ for elements x and y of an n-element partial order P. As we shall see later, this yields a fpras for Linear

Extension Count. First, we give a brief sketch of the Karzanov-Khachiyan algorithm.

Theorem 3 (Karzanov-Khachiyan). *There is a randomized algorithm with the following properties. The input is an n-element partial order P and a positive number ϵ. The output is a linear extension of P, and for any $\lambda \in \Lambda(P)$ we have*

$$\left|\Pr(\lambda \text{ is output}) - \frac{1}{N(P)}\right| < \frac{\epsilon}{N(P)}.$$

The running time of the algorithm is of order $n^6 \log n \log(1/\epsilon)$.

Sketch of Proof. Let P be an n-element partial order. We define the *linear extension graph* $G(P)$ of P to be a graph with vertex set $\Lambda(P)$ and two linear extensions λ, μ adjacent if they differ by an adjacent transposition. So the degree $d(\lambda)$ of a vertex of $G(P)$ is at most $n-1$. We consider the following random walk, whose state space is $\Lambda(P)$, and whose transition matrix is given by:

$$p(\lambda, \mu) = \begin{cases} 1/(2n-2) & \lambda \text{ and } \mu \text{ are adjacent,} \\ 1 - d(\lambda)/(2n-2) & \lambda = \mu, \\ 0 & \text{otherwise.} \end{cases}$$

Note in particular that the chain stays in the same state with probability at least $1/2$. It is evident that this chain is strongly connected and aperiodic, and that its unique stationary distribution is given by $\pi(\lambda) = 1/N(P)$ for every $\lambda \in \Lambda(P)$. Let $\pi(\lambda, t)$ denote the probability that the chain is in state λ after t steps, given some initial distribution. Then

$$\pi(\lambda, t) \to \pi(\lambda) = 1/N(P),$$

as t tends to infinity, independent of the initial distribution. A Markov chain is said to be *rapidly mixing* if, roughly speaking, this convergence takes place in time polynomial in n.

The *conductance* α of the graph $G(P)$ is defined as

$$\alpha = \frac{1}{2n-2} \min_X \left\{ \frac{|E(X, \overline{X})|}{|X|} \right\},$$

where the minimum is taken over all subsets X of $\Lambda(P)$ with $1 \le |X| \le N(P)/2$, and $E(X, \overline{X})$ is the number of edges from X to its complement $\overline{X}$ in $G(P)$. The relevance of this parameter is that, if the graph has small conductance, the Markov chain may become 'trapped' in X. Thus we would not expect rapid mixing to take place if the conductance is too small. A result of Sinclair and Jerrum [24] states that

$$\left|\pi(\lambda, t) - \frac{1}{N(P)}\right| \leq \left(1 - \frac{\alpha^2}{2}\right)^t,$$

for all $\lambda \in N(P)$, regardless of the initial distribution.

Now it can be shown, using geometric arguments about convex bodies, that $\alpha > 2^{-3/2}n^{-5/2}$. See Karzanov and Khachiyan [16] or Lovász and Simonovits [21]. Combining this with the Sinclair-Jerrum result, we find that

$$|\pi(\lambda, t) - 1/N(P)| < \exp(-t/16n^5) < \epsilon/N(P),$$

provided t is at least $16n^5 \log(N(P)/\epsilon) \leq 16n^6 \log n \log(1/\epsilon)$.

The algorithm we require runs as follows. Some linear extension of P is found, and becomes the initial point of the Markov chain. The Markov chain is then run for $T = 16n^6 \log n \log(1/\epsilon)$ steps, producing a random linear extension with the required distribution properties.

The running time estimate we give is simply $O(T)$. This ignores the time required to generate a random integer in the range $[1, 2n-2]$. $\square$

Our intention is to estimate $\Pr(x \prec y \mid P)$, for P an n-element partial order. It turns out that we want to evaluate this to within a multiplicative constant of $(1+\eta)$. If $\Pr(x \prec y)$ is exponentially small in n, this is asking too much, but it will be enough to estimate the probability under the assumption that $\Pr(x \prec y) > 2/5$. It is clear what to do: we run the Karzanov-Khachiyan algorithm a large number of times, and take the proportion of generated linear extensions in which x precedes y as our estimate for $\Pr(x \prec y)$.

Theorem 4. *Let $0 < \eta, \delta < 1/3$ be given. There is an algorithm which, presented with an n-element partial order P and an ordered pair (x, y) of incomparable elements of P with*

Pr$(x \prec y \mid P) = \gamma > 2/5$*, outputs an estimate* U *for* γ *such that*

$$\Pr\left(\left|\frac{U}{\gamma} - 1\right| > \eta\right) < \delta.$$

The algorithm runs in time $O(n^6 \log n \log(1/\eta)\eta^{-2} \log(1/\delta))$.

Proof. We set $\epsilon = \eta/3$, and apply Theorem 3. The probability p that one run of the Karzanov-Khachiyan algorithm produces a linear extension with $x \prec y$ satisfies $|p/\gamma - 1| \leq \epsilon$. If we perform N runs, the number of linear extensions observed with $x \prec y$ is a Binomial random variable $S_{N,p}$ with parameters N and p. A standard calculation gives that

$$\Pr(|S_{N,p} - Np| > \epsilon Np) < \exp(-\epsilon^2 N/10)$$

whenever $p \geq 1/3$.

The conclusion is that the proportion U of the N randomly generated linear extensions with $x \prec y$, distributed as $S_{N,p}/N$, is a good approximation to p, and hence to γ. Indeed $|\gamma - p| < \epsilon\gamma < \epsilon(3\gamma - p)$, and so

$$\begin{aligned} \Pr(|U - \gamma| > 3\epsilon\gamma) &\leq \Pr(|U - p| > \epsilon p) \\ &< \exp(-\epsilon^2 N/100). \end{aligned}$$

This can be converted to the form required by replacing 3ϵ by η, and setting $N = 100\eta^{-2}\log(1/\delta)$. The time estimate given is N times the number of steps of the chain required to produce one approximately random linear extension. □

Finally, let us see how this enables us to approximate $N(P)$.

Theorem 5. *There is a randomized algorithm with the following properties. The input is an* n*-dimensional partial order* P*, and positive rationals* ϵ, β*. The output is a number* L *such that*

$$\Pr\left(\left|\frac{L}{N(P)} - 1\right| > \epsilon\right) < \beta.$$

The running time of this algorithm, is $O(n^9 \log^6 n \log(1/\epsilon)\epsilon^{-2} \log(1/\beta))$.

Note that Theorem 5 is the same as Theorem 2, with an explicit estimate for the running time.

Proof. The main idea is to find a sequence of n-vertex posets $P = P_0, P_1, \dots, P_k$, where P_k is a linear order, such that each P_{j+1} is obtained from P_j by adding one new relation $a_j < b_j$ (and taking the transitive closure). Then we can write

$$N(P) = \prod_{j=0}^{k-1} \frac{N(P_j)}{N(P_{j+1})} = \prod_{j=0}^{k-1} \left(\Pr(a_j \prec b_j \mid P_j)\right)^{-1}.$$

If we arrange matters so that all the terms $\Pr(a_j \prec b_j \mid P_j)$ are at least $2/5$, then we can apply Theorem 4 to estimate each term. Roughly speaking, it is enough to estimate the probabilities to within a multiplicative factor of $(1+\epsilon/k)$, with probability of error at most β/k. As we also have to make k estimates, it can be seen that the running time estimate contains a factor k^3, so it is worth taking a little effort to minimize k. It is fairly easy to achieve $k \le n^2$, but one can actually get $k = O(n \log n)$.

Given an n-vertex poset P, we define our sequence (P_j) of posets as follows. The initial poset P_0 is just P. We now follow some comparison sorting algorithm that uses at most $2n \log n$ comparisons in the worst case, such as binary insertion sorting. When the algorithm calls for a comparison to be made, we do the following. If the two elements to be compared are already related in the current P_j, we adopt this relation as the result of our comparison. If the pair (a, b) is not related in P_j, we run the algorithm of Theorem 4 to estimate $\Pr(a \prec b \mid P_j)$, with $\eta = \epsilon/(4n \log n)$, and $\delta = \beta/2n \log n$. Without loss of generality, the estimate E_j we obtain for $\Pr(a \prec b)$ is at least $1/2$. Therefore, with probability at least $1 - \delta$, the true probability is at least $2/5$, and

$$\left| \frac{E_j}{\Pr(a \prec b \mid P_j)} - 1 \right| < \eta.$$

In this case, we set $a_j = a$ and $b_j = b$, add the relation $a_j < b_j$ to P_j to form P_{j+1}, and return $a < b$ as the result of the comparison.

At each stage, P_j contains all the relations known to the sorting algorithm, so the process terminates with P_k a linear order for $k \leq 2n \log n$.

We now take $L = \prod_{j=0}^{k-1} E_j^{-1}$ as our estimate for $N(P)$, so that

$$\frac{L}{N(P)} = \prod_{j=0}^{k-1} \frac{\Pr(a_j \prec b_j \mid P_j)}{E_j}.$$

With probability at least $1 - k\delta \geq 1 - \beta$, all the E_j are within the stated bounds, so this is a product of k terms lying between $(1 + \epsilon/2k)^{-1}$ and $(1 - \epsilon/2k)^{-1}$. Therefore the estimate L satisfies

$$1 - \epsilon < \left(\frac{1}{1 + \epsilon/2k}\right)^k < \frac{L}{N(P)} < \left(\frac{1}{1 - \epsilon/2k}\right)^k < 1 + \epsilon,$$

with probability at least $1 - \beta$, as desired.

The running time of this process is dominated by the $k \leq 2n \log n$ estimates of $\Pr(a \prec b)$. Each of these takes time of order

$$n^6 \log n \log(1/\eta)\eta^{-2} \log(1/\delta) \leq C n^8 \log^5 n \log(1/\epsilon)\epsilon^{-2} \log(1/\beta),$$

so the total running time is as given. $\square$

In practice, the algorithm can doubtless be made to run somewhat faster, since the rapid mixing of the Markov chain usually takes place in time considerably less than that given in Theorem 3.

4. Proof of Theorem 1.

In this section, we show that Linear Extension Count is #P-complete. Before going on to the formal proof, we give a brief outline. We suppose that we are given an instance I of 3-SAT Count with m variables and n clauses, and that we have an oracle which returns the number of linear extensions of any partially ordered set P of size at most some polynomial in n and m. Our first step is to construct from our instance I an auxiliary poset P_I of size $7n + m$, and use our oracle to calculate the number L_I of linear extensions of P_I. Next, we find a set S of primes between $7n + m$ and $(7n + m)^2$, whose product is at least 2^m, such that no prime in S divides L_I. Our aim is now to find the number of satisfying assignments of I, mod p, for each prime $p \in S$. Since the number of satisfying assignments is at most 2^m, this will determine the number of satisfying assignments for I.

For each prime $p \in S$, we now form a poset $Q_I(p)$ of size about $p(n + m)$, with the property that the number of linear extensions of $Q_I(p)$ can be written as $\alpha p + s(I)\beta\gamma L_I$, where α is a positive integer, β and γ are easily computable integers (depending on p, n and m) neither of which is divisible by p, and $s(I)$ is the number of satisfying assignments of I. Then we use our oracle to compute $L(Q_I(p))$, which is equal to $s(I)\beta\gamma L_I$, mod p. Now we are able to find $s(I)$ mod p, as desired.

Proof of Theorem 1. Suppose we have an oracle $\mathcal{O}(t)$ which, when presented with a partially ordered set P of size at most t, returns in unit time the number of linear extensions of P.

We shall give an algorithm which solves the problem 3-SAT Count in time polynomial in the number m of variables and the number n of clauses, making use of the oracle $\mathcal{O}(t)$, where $t = (7n + m)^3$.

Thus, let I be an instance of 3-SAT Count, consisting of m variables and n clauses which are conjunctions of three literals. For convenience, we set $M = 7n + m$.

Let P_I be the partially ordered set defined from I as follows. The points of P_I

consist of a vertex h_x corresponding to each variable x in the instance, and seven vertices for each clause. If x, y and z are the variables in some particular clause, then each of the seven vertices corresponding to that clause is placed above a different non-empty subset of $\{h_x, h_y, h_z\}$. There are no other comparabilities in P_I. (See Figure 1.)

Figure 1. The relations in P_I corresponding to a clause involving the variables x, y and z.

Let L_I be the number of linear extensions of P_I. Since the size of P_I is just M, this number can certainly be calculated using $\mathcal{O}(M^3)$.

Let S_0 be the set of primes strictly between M and M^2. By the number-theoretic lemma given in Section 2, their product is at least $M!\,2^m$. Since L_I is at most $M!$, there is a set S of primes strictly between M and M^2, none of which divide L_I, whose product is at least 2^m.

Let p be a prime in S. We now define a partially ordered set $Q_I(p)$ as follows. (See Figure 2.)

Figure 2. The poset $Q_I(p)$. Here the ovals represent antichains of size $p-1$. The only clause vertices shown here are those corresponding to the clause $xy\overline{z}$.

There are two special vertices a and b which are used to divide linear extensions of the poset into three sections. The section below a will be referred to as the *bottom* section; that between a and b is the *middle* section, and the section above b is the *top*. Some of the other vertices will be bound into one particular section; others will be free to appear in different sections, depending on the linear extension.

Below a in $Q_I(p)$ is an antichain U of size $(m+1)(p-1)$. This antichain is divided into sets of size $p-1$, one set U_x corresponding to each variable x in the instance, and one extra set U_0.

Similarly, between a and b is an antichain V of size $(n+1)(p-1)$. Again this is divided into sets of size $p-1$, with a set V_c corresponding to each clause c, and one other set V_0.

Next, for each variable x in the instance, we have two corresponding *literal vertices* which we shall refer to as x and $\overline{x}$. (Thus we abuse notation by using the same symbol to represent both a literal and a vertex of the poset.) The literal vertices x and $\overline{x}$ are incomparable with both a and b, and are above all elements in the set U_x corresponding to the variable x.

Finally, we have eight *clause vertices* $c_1, c_2, \ldots, c_8$ for each clause c of the instance. If x, y and z are the three variables involved in the clause c, then there is a clause vertex above each triple of literal vertices consisting of one element from each of $\{x, \overline{x}\}$, $\{y, \overline{y}\}$, $\{z, \overline{z}\}$. The clause vertex c_i which is above that triple of literals which actually constitutes the clause c is also above b; the other clause vertices are above each element of the antichain V_c corresponding to c. Thus all clause vertices are above a.

The total number of vertices in the poset $Q_I(p)$ is thus

$$2 + (p-1)(n+m+2) + 2m + 8n < p(7n+m) \leq M^3.$$

So the number of linear extensions of $Q_I(p)$ can be found using the oracle $\mathcal{O}(M^3)$. We next investigate how this number is related to the number of satisfying assignments for I.

We shall partition the set of linear extensions of $Q_I(p)$ according to which of the literal and clause variables occur in each of the three sections marked off by a and b.

We define a *configuration* to be a partition ϕ of the literal and clause vertices into three sets B^ϕ, M^ϕ and T^ϕ. Let Φ denote the set of all configurations. We say a linear extension of $Q_I(p)$ *respects* a configuration $\phi = (B^\phi, M^\phi, T^\phi)$ if $B^\phi \prec a \prec M^\phi \prec b \prec T^\phi$ in the linear extension. The set of linear extensions respecting a configuration ϕ is denoted L^ϕ.

We say that a configuration is *consistent* if L^ϕ is non-empty, which is the case whenever the information $B^\phi \prec a \prec M^\phi \prec b \prec T^\phi$ is consistent with the partial order $Q_I(p)$. Also, if L^ϕ is non-empty, it is just the set of linear extensions of the partial order P^ϕ defined by adding to $Q_I(p)$ the relations given by $B^\phi < a < M^\phi < b < T^\phi$ and taking the transitive closure.

Thus we have

$$N(Q_I(p)) = \sum_{\phi \in \Phi} N(P^\phi).$$

We shall prove that the only configurations which contribute to this sum, mod p, are those where B^ϕ contains exactly one literal vertex for each variable, M^ϕ contains exactly one clause vertex for each clause, and T^ϕ contains the remaining literal and clause vertices. Furthermore, this is only possible when the set of literal vertices in T^ϕ corresponds to a satisfying assignment for I, and each satisfying assignment gives rise to exactly one such consistent configuration. Finally, when $N(P^\phi)$ is not divisible by p, it is equal to a readily calculable constant.

As a first step towards proving these assertions, let us remark that, for any consistent configuration ϕ, the vertices a and b are comparable with every other vertex in P^ϕ. Let P^ϕ_B be the poset induced on the elements below a in P^ϕ, P^ϕ_M the poset induced on the elements between a and b, and P^ϕ_T the poset induced on the elements above b. Now we have

$$N(P^\phi) = N(P_B^\phi)\, N(P_M^\phi)\, N(P_T^\phi).$$

Thus $N(P^\phi)$ is divisible by p precisely when one of these three terms is.

A consistent configuration ϕ is said to be *feasible* if neither $N(P_B^\phi)$ nor $N(P_M^\phi)$ is divisible by p.

Let ϕ be any feasible configuration. We consider first the bottom section P_B^ϕ of the poset P^ϕ. This consists of the antichain U of size $(p-1)(m+1)$, together with some of the literal vertices. The elements of U_0 are isolated in this poset, as are the elements of U_x for any x such that neither of the two associated literal vertices x and $\overline{x}$ is in B^ϕ. Let $k = |P_B^\phi|$, and let $r \geq p-1$ be the number of isolated vertices.

A linear extension of P_B^ϕ can be considered as a choice of positions among the heights $1, 2, \ldots, k$ for each of the r isolated vertices, together with a linear extension of the poset induced on the remaining vertices. Hence $N(P_B^\phi)$ is divisible by $k(k-1)\ldots(k-r+1)$. Since ϕ is feasible, this quantity is not divisible by p, and so $r = p-1$, and $k \equiv -1 \pmod{p}$. Since k lies between $(p-1)(m+1)$ and $(p-1)(m+1)+2m$, and $p > m$, this implies that exactly m literal vertices are in B^ϕ—one for each variable.

Therefore the poset P_B^ϕ consists of $p-1$ isolated elements, and m components consisting of one literal vertex above an antichain of size $p-1$. We claim that the number of linear extensions of a poset P of this form is exactly

$$\big(p(m+1)-1\big)!/p^m.$$

To see this, for each variable x, let A_x be the event that a randomly chosen ordering of the vertices of P has the literal vertex associated with x above all "its" $p-1$ vertices. The probability of each A_x is $1/p$, and the m events are independent. Moreover, an ordering of the vertices is a linear extension of P iff each A_x occurs.

The above product $\big(p(m+1)-1\big)!$ has just m terms which are multiples of p, and none which are mutiples of higher powers of p, so $\big(p(m+1)-1\big)!/p^m$ is not divisible by p.

To summarize, if ϕ is feasible, then B^ϕ contains exactly one literal vertex for each variable, and $N(P_B^\phi) = \big(p(m+1)-1\big)!/p^m \not\equiv 0 \pmod{p}$.

We now move up and consider the middle section of P^ϕ. The argument in this case is essentially identical to that for the bottom section.

We assume once more that the configuration ϕ is feasible. We are now concerned with the middle section P_M^ϕ of the poset P^ϕ. This consists of the antichain V of size $(p-1)(n+1)$, together with some of the literal and clause variables, say j of them. Note that $0 \le j \le 7n+m < p$. Each of the $p-1$ elements of V_0 is isolated in P_M^ϕ, as are all the elements of V_c for any clause c, none of whose associated clause vertices c_i are in M^ϕ.

Arguing exactly as for the bottom section, we see that, for each clause c, at least one of the vertices c_i associated with c appears in M^ϕ, and that the total number of vertices in P_M^ϕ is congruent to $-1 \pmod{p}$. The only possibility is that exactly n of the literal and clause vertices are in the middle section. Thus M^ϕ contains no literal vertices and exactly one clause vertex for each clause.

Again essentially as for the bottom case, we have that, if P_M^ϕ is of this form, then

$$N(P_M^\phi) = \big(p(n+1)-1\big)!/p^n \not\equiv 0 \pmod{p}.$$

We know that, in each feasible configuration, every variable x has one of its associated literal vertices l_x appearing in B^ϕ, and the other, h_x, in T^ϕ. Thus each feasible configuration induces an assignment $h(\phi)$ of true literals for the instance I, consisting of the literals h_x. We shall show that an assignment that satisfies the instance corresponds to just one feasible configuration, whereas an assignment that does not satisfy the instance corresponds to no feasible configurations. This will imply that the number of feasible configurations is equal to the number of satisfying assignments.

Suppose ϕ is feasible, and let c be a clause involving variables x, y and z. Then seven of the eight associated clause vertices c_i are above at least one of h_x, h_y or h_z, and are therefore themselves in T^ϕ. Therefore it is the eighth clause vertex which appears in M^ϕ, namely that c_i whose designated triple of literal vertices is $\{l_x, l_y, l_z\}$. Therefore the assignment $h(\phi)$ determines ϕ uniquely. If, however, the set $\{l_x, l_y, l_z\}$ corresponds exactly to the set of literals in the clause c, then this chosen clause vertex is above b in $Q_I(p)$, and therfore is necessarily in T^ϕ.

In other words, if any clause is not satisfied by the assignment $h(\phi)$, then ϕ is not feasible, a contradiction. Conversely, if h is any satisfying assignment, then $h = h(\phi)$ for some feasible ϕ, namely the configuration where B^ϕ consists of the literal vertices corresponding to false literals, and M^ϕ consists of the clause vertices which are above only "false" literal vertices.

The next observation is that, if ϕ is a feasible configuration, then the poset P_T^ϕ is isomorphic to the auxiliary poset P_I. Indeed, each variable x is represented by the literal h_x in T^ϕ, and each clause by seven of the eight associated clause vertices. If x, y and z are the variables involved in a clause c, then every non-empty subset of $\{h_x, h_y, h_z\}$ has one clause vertex c_i above just the elements of that subset.

We are now in a position to count the number of linear extensions of $Q_I(p)$, mod p. We know that non-feasible configurations contribute nothing to this sum, and feasible configurations are in 1–1 correspondence with satisfying assignments for I. Moreover, for each feasible configuration ϕ,

$$\begin{aligned} N(P^\phi) &= N(P_B^\phi)\, N(P_M^\phi)\, N(P_T^\phi) \\ &= \big(p(m+1)-1\big)!/p^m \cdot \big(p(n+1)-1\big)!/p^n \cdot L_I, \end{aligned}$$

and none of the three terms making up this product, which we denote by N_0, is divisible by p. (In the case of L_I, this is by definition of the set S of primes we are using.)

In other words, for each feasible configuration ϕ, $N(P^\phi)$ is equal to some N_0 depending on p, n and m but not ϕ. Therefore

$$N(Q_I(p)) \equiv N_0 \cdot s(I) \pmod{p}.$$

Furthermore, N_0 is not divisible by p, and can be calculated quickly.

The oracle $\mathcal{O}(M^3)$ enables us to find the number of linear extensions of $Q_I(p)$ for each prime p in our set S. This then enables us to find $s(I) \pmod p$ for every $p \in S$. Since the product of the primes in S is greater than 2^m, and $s(I)$ is at most 2^m, we can then find the value of $s(I)$.

It remains to check that this procedure is polynomial, given the oracle $\mathcal{O}(M^3)$, where $M = 7n + m$. Much of the procedure consists of arithmetic manipulation, and the largest number we have to deal with is at most $(M^3)!$, an overestimate for the number of linear extensions of any $Q_I(p)$. Thus, at the cost of a factor of, say, M^6, we may assume that all the arithmetic operations we carry out take unit time.

The number of calls to the oracle is $|S|+1 < M^2$. Setting up the poset P_I takes time $O(M^2)$. Finding the set S_0 of primes can be done by a sieve in time $O(M^2)$, and removing those which divide L_I by trial division takes time $O(M)$. The number of primes in S is at most M^2, and for each of those we construct a poset $Q_I(p)$ of size at most M^3. Setting up the poset for submission to the oracle thus takes time at most $O(M^6)$. Calculating the quantities $\big(p(m+1)-1\big)!/p^m$, $\big(p(n+1)-1\big)!/p^n$, and L_I, all mod p, takes time $O(M^3)$, and inverting their product N_0 mod p can certainly be done in time $O(M^2)$. Given this inverse, and $L(Q_I(p))$, mod p, we can calculate $s(I)$, mod p, in unit time. Finally, given all the at most M^2 values of $s(I)$, mod p, for every $p \in S$, we can find $s(I)$ in time $O(M^3)$. The whole procedure thus has time complexity $O(M^{14})$, and in fact this can easily be improved to about $O(M^5)$ by using a sharper version of our number-theoretic lemma and a more careful analysis.

This completes the proof of Theorem 1. $\square$

Let us make a few remarks about the above proof. Firstly, it is known to be #P-complete to compute the number of satisfying assignments for a Boolean formula in 2-conjunctive normal form, even under very restrictive conditions: see [19,23]. We chose to reduce from 3-SAT Count for reasons of familiarity, as there would be no significant simplification of the proof obtained from using 2-SAT instead.

Note that our construction proves # P-completeness for Linear Extension Count for posets of height at most 5. In fact, we can alter the construction slightly so as to get the height down to 3, as we describe below.

We form a poset $Q'_I(p)$ from $Q_I(p)$ by removing all the comparabilities between $U \cup \{a\}$ and $V \cup \{b\}$ (keeping a below all the clause variables). Again, we partition the linear extensions of $Q'_I(p)$ according to the set B^ϕ of elements coming below a. If b comes below a, then the number of linear extensions of the poset restricted to B^ϕ is divisible by p, since the entire set V forms an antichain of indistinguishable elements, and so the number of linear extensions is divisible by $|V|!$. Similarly, if $b \notin B^\phi$, but some element v of V is, then v together with U_0 forms an antichain of size p below a, so again the number of linear extensions of the poset restricted to B^ϕ is divisible by p. Hence the number of linear extensions of $Q'_I(p)$ is congruent to the number of linear extensions of $Q_I(p) (\mathrm{mod}\ p)$, and so, to solve the instance I of 3-SAT Count, it is sufficient to be able to count the linear extensions of the posets P_I and $Q'_I(p)$, all of which have height at most 3.

We strongly suspect that Linear Extension Count for posets of height 2 is still #P-complete, but it seems that an entirely different construction is required to prove this.

5. Related Problems.

We now discuss the implications of Theorem 1 for some closely related problems.

We saw in Section 3 that the number of linear extensions of a poset P was related to the volume of the order polytope $Q(P)$ by the simple formula $\mathrm{vol}(Q(P)) = N(P)/n!$. Thus Theorem 1 implies that it is #P-complete to evaluate $\mathrm{vol}(Q(P))$. Yet $Q(P)$ is simply the intersection of at most n^2 half-spaces specified by inequalities of the form $x_i - x_j < 0$. Therefore the problem of calculating the volume of an n-dimensional polytope is strongly #P-complete. This consequence of our Theorem 1 was first pointed out by Khachiyan [17].

We next consider the problem of evaluating $\Pr(x \prec y)$. Strictly speaking, this problem does not belong to the class $\#P$, as it is not an enumeration problem. However, in view of the following theorem, it may effectively be regarded as a #P-complete problem.

Theorem 6. *The problem of evaluating $Pr(x \prec y)$ in a poset P is polynomially equivalent to Linear Extension Count.*

Proof. Given an oracle for Linear Extension Count, we can apply this to the two posets P and $P \cup (x, y)$, and derive $\Pr(x \prec y \mid P)$, so evaluating $\Pr(x \prec y)$ is certainly no harder than Linear Extension Count.

For the converse, we can use the method of Theorem 5. Let P be an instance of Linear Extension Count with n vertices, and suppose we have an oracle which computes $\Pr(x \prec y \mid Q)$ in unit time whenever x and y are incomparable elements of an n-element poset Q.

As in Theorem 5, we can find, in time polynomial in n, a sequence of posets (P_j) and relations (a_j, b_j) so that

$$N(P) = \prod_{j=0}^{k-1} \left(\Pr(a_j \prec b_j \mid P_j)\right)^{-1},$$

with $k \leq 2n \log n$. Our oracle can then evaluate these probabilities, which are rational numbers with denominator at most $n!$, and $N(P)$ can be calculated from these. $\square$

A second related problem is that of determining the average height of a vertex in a poset. If x is a vertex in a poset P, and $\prec$ is a linear extension of P, then the *height* of x in $\prec$ is the number of elements below x in $\prec$, plus one. The *average height* $H_P(x)$ of x in P is the average over all linear extensions $\prec$ of P of the height of x in $\prec$.

Theorem 7. *The problem of determining the average height of an element of a poset is polynomially equivalent to that of evaluating* $\Pr(x \prec y)$*.*

Proof. We have the following two identities:

$$H_P(x) = \sum_{y \neq x} \Pr(y \prec x \mid P) + 1,$$

$$H_P(x) = (1 - \Pr(x \prec y \mid P))\, H_{P \cup (y,x)}(x) + \Pr(x \prec y \mid P)\, H_{P \cup (x,y)}(x).$$

The first identity enables us to calculate $H_P(x)$ given an oracle for $\Pr(x \prec y)$, and the second allows us to compute $\Pr(x \prec y)$ given an oracle for average height. $\square$

Thus the problem of determining the average height of an element in a poset is also seen to be polynomially equivalent to a #P-complete problem.

References.

[1] D. Applegate and R. Kannan, Sampling and integration of near log-concave functions, *Proc. 23rd ACM Symposium on the Theory of Computing* (1991) 156–163.

[2] M.D. Atkinson, Partial orders and comparison problems, *Congressus Numerantium* **47** (1985) 77–88.

[3] M.D. Atkinson and H.W. Chang, Extensions of partial orders of bounded width, *Congressus Numerantium* **52** (1985) 21–35.

[4] M.D. Atkinson and H.W. Chang, Computing the number of mergings with constraints, *Information Processing Letters* **24** (1987) 289–292.

[5] G. Brightwell and P. Winkler, Counting linear extensions is #P-complete, *Proc. 23rd ACM Symposium on the Theory of Computing* (1991) 175–181.

[6] A.Z. Broder, How hard is it to marry at random? (On the approximation of the permanent), *Proc 18th ACM Symposium on the Theory of Computing* (1986) 50–58.

[7] M. Dyer and A. Frieze, On the complexity of computing the volume of a polyhedron, *SIAM J. Computing*, to appear.

[8] M. Dyer and A. Frieze, Computing the volume of convex bodies: a case where randomness provably helps, preprint.

[9] M. Dyer, A. Frieze and R. Kannan, A random polynomial time algorithm for estimating volumes of convex bodies, *Proc. 21st ACM Symposium on the Theory of Computing* (1989) 375–381.

[10] J. Feigenbaum, private communication.

[11] P.C. Fishburn and W.V. Gehrlein, A comparative analysis of methods for constructing weak orders from partial orders, *J. Math. Sociology* **4** (1975) 93–102.

[12] M. Habib and R. H. Mohring, On some complexity properties of N-free posets and posets with bounded decomposition diameter, *Discrete Math.* **63** (1987) 157–182.

[13] G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers*, 4th Ed., Oxford University Press 1960.

[14] M. Jerrum and A. Sinclair, Conductance and the rapid mixing property for Markov chains: the approximation of the permanent resolved, *Proceedings of the 20th ACM Symosium on Theory of Computing* (1988) 235–244.

[15] J. Kahn and M. Saks, Balancing poset extensions, *ORDER* **1** No. 2 (1984) 113–126.

[16] A. Karzanov and L. Khachiyan, On the conductance of order Markov chains, Technical Report DCS TR 268, Rutgers University, June 1990.

[17] L. Khachiyan, Complexity of polytope volume computation, *Recent Progress in Discrete Computational Geometry*, J. Pach ed., Springer-Verlag, to appear.

[18] H. Kierstead and W.T. Trotter, The number of depth-first searches of an ordered set, *ORDER*, to appear.

[19] N. Linial, Hard enumeration problems in geometry and combinatorics, *SIAM J. Alg. Disc. Meth.* **7** No. 2 (1986) 331–335.

[20] L. Lovász, *An Algorithmic Theory of Numbers, Graphs and Convexity*, SIAM, Philadelphia 1986.

[21] L. Lovász and M. Simonovits, The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume, *Proc. 31st IEEE Symposium on Foundations of Computer Science* (1990) 346–355.

[22] P. Matthews, Generating a random linear extension of a partial order, *Annals of Probability*, to appear.

[23] S. Provan and M.O. Ball, On the complexity of counting cuts and of computing the probability that a graph is connected, *SIAM J. Computing* **12** (1983) 777–788.

[24] A. Sinclair and M. Jerrum, Apporximate counting, generation and rapidly mixing Markov chains, *Information and Computation* **82** (1989) 93–133.

[25] G. Steiner, Polynomial algorithms to count linear extensions in certain posets, *Congressus Numerantium*, to appear.

[26] G. Steiner, On counting constrained depth-first linear extensions of ordered sets, preprint.

[27] S. Toda, On the computational power of PP and +P, *Proc. 30th IEEE Symposium on*

Foundations of Computer Science (1989) 514–519.

[28] L.G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* **8** (1979) 189–201.

[29] L.G. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* **8** (1979) 410–421.

[30] P. Winkler, Average height in a partially ordered set, *Discrete Math.* **39** (1982) 337–341.