Randomised Algorithms for Counting and Generating Combinatorial Structures

Alistair Sinclair

Ph. D.

University of Edinburgh

1988

Abstract

The thesis studies the computational complexity of two natural classes of combinatorial problems: counting the elements of a finite set of structures and generating them uniformly at random. For each problem class, a notion of efficient randomised approximation algorithm is defined. The central theme is the classification of counting and generation problems for natural combinatorial structures with respect to these notions of tractability. For many structures, the two problems are of essentially equivalent complexity at this level of approximation.

The emphasis is on positive results. In particular, the thesis makes a detailed study of a general technique for random generation based on simulating a finite Markov chain whose states are the structures of interest and which converges to some desired distribution over them. The efficiency of this method depends crucially on the rate of convergence of the chain. A major portion of the thesis is devoted to deriving a characterisation of rapid convergence for a broad class of Markov chains in terms of a structural property of the underlying graph. This enables useful bounds on the rate of convergence of non-trivial chains to be established for the first time. As a result, the following classes of structures may be generated almost uniformly in polynomial time: perfect matchings in dense graphs, and matchings of all sizes in arbitrary graphs. This approach also yields the first polynomial time approximation algorithms for counting these structures, and hence for computing the permanent of dense 0-1 matrices and the partition function of monomer-dimer systems in statistical physics.

It is also shown that, for self-reducible structures, almost uniform generation is polynomial time reducible to a very weak form of approximate counting. This fact yields the following robustness result for approximate counting: for self-reducible structures, polynomial time randomised algorithms for counting to within factors of the form $1 + n^{\beta}$ are available either for all real β or for no real β. It also provides a mechanism for generating structures given fairly crude analytic counting estimates for them. This technique is used to derive an almost uniform generation procedure for labelled graphs with given degree sequence and a given excluded subgraph which is valid over a much wider range of degrees than previous methods. This in turn leads to approximate counting algorithms for these graphs with very good asymptotic behaviour.

Contents

Synopsis

This thesis is concerned with two classes of problems involving a finite set S of combinatorial structures: counting them and generating them randomly from a uniform distribution. The set S will always have a concise implicit description x: for example, x may be a connected graph and S its set of spanning trees, or x a partial order and S its set of linear extensions. We also consider the more general setting in which each structure in S has an associated positive weight: the task is then to compute the weighted sum of all elements of S, or to generate them randomly with probabilities proportional to their weights.

Combinatorial counting problems have a long and distinguished history which we will not attempt to summarise here. In addition to their intrinsic interest, they arise naturally from investigations in numerous other branches of mathematics and the natural sciences and have given rise to a rich and beautiful theory.

Generation problems are less well studied but have a number of computational applications. For example, uniform generation can be seen as a way of exploring a large set of combinatorial structures and constructing typical representatives of it. These may be used to formulate conjectures about the set, or perhaps as test data for the empirical analysis of some heuristic algorithm which takes inputs from the set. Non-uniform generation occurs in the mathematical modelling of physical systems where the structures are valid system configurations each of which has a weight which depends on its energy. Many important properties of the model can be deduced from estimates of the expectation, under the weighted distribution, of certain operators on configurations. An analogous idea lies at the heart of currently fashionable stochastic techniques for combinatorial optimisation, such as simulated annealing. Here, low cost solutions have low "energy" and thus large weight: sampling from the weighted distribution

therefore tends to favour them.

The central question we shall address is that of the existence of computationally efficient counting and generation procedures for a given class of structures. In accordance with accepted practice in theoretical computer science, "efficient" is taken to mean having a runtime which grows only polynomially with the size of the problem description x. Of course, for most structures this rules out the naïve approach of exhaustively enumerating the elements of S since there will in general be far too many of them.

Our treatment is motivated by the empirical observation that efficient *exact* counting procedures exist only for a relatively small number of interesting structures. For many others, the apparent hardness of counting is supported by strong theoretical evidence to the effect that no efficient counting procedure exists unless P = NP. However, there is no *a priori* reason to suppose that some of these structures cannot be counted *approximately* in some suitable sense. Specifically, it is interesting to relax the requirements of exact counting in two directions: randomisation and approximation. Thus we allow our counting procedures to flip coins, and demand that they produce an answer which is correct to within some small specified error with high probability. In the case of generation, we settle for a distribution which is almost uniform (or, more generally, close to some weighted distribution) with some small specified bias. This gives us well-defined notions of approximate counting and almost uniform generation: a fundamental assumption implicit in our approach is that they correspond to reasonable notions of effective tractability.

The principal aim of this thesis is to go some way towards classifying naturally occurring counting and generation problems according to these notions of tractability. We shall concentrate mainly on techniques for proving positive results, though it is worth pointing out that for certain structures the problems remain hard even in approximate form. Our investigation will have essentially two strands: the classification of particular structures, and the nature of the classification process itself.

The reader may be wondering why we have chosen to investigate two apparently very disparate classes of problems at the same time. The reason for this is that, from a computational point of view, the problems of approximate counting and almost uniform generation are very closely related. More precisely, for most

natural structures a polynomial time procedure for approximate counting can be used to construct a polynomial time almost uniform generation algorithm, and *vice versa*. The only assumption we need to make is that the structures are *self-reducible*, which essentially means that they possess a simple inductive construction in terms of similar structures of a smaller size. This connection has often been observed in specific cases, and was recently formalised in a general setting by Jerrum, Valiant and Vazirani [33], whose work initially inspired the author to embark on the research presented in this thesis. The upshot is that counting and generation problems may profitably be studied in tandem: techniques which naturally suggest themselves for one class of problems can be used to solve the other indirectly.

Traditionally the cross-fertilization between these problems has tended to be rather one-sided: exact or approximate analytic results on counting enable one to generate structures uniformly or almost uniformly. This thesis concentrates chiefly on the opposite direction. We analyse a powerful general tool for generation, which by virtue of self-reducibility constitutes an attack on counting problems as well.

The tool in question, which has been in use for some years in statistical physics under the name of the Monte Carlo method [10], involves constructing a dynamic stochastic process (a finite Markov chain) whose states correspond to the set S of structures of interest. The process is able to move around the space by means of random local perturbations of the structures. Moreover, the process is *ergodic*, i.e., if it is allowed to evolve in time the distribution of the final state tends asymptotically to a unique *stationary distribution* π over S which is independent of the initial state. By simulating the process for sufficiently many steps and outputting the final state, we are therefore able to generate elements of S from a distribution which is arbitrarily close to π.

The construction and simulation of an ergodic Markov chain with the desired stationary distribution is usually straightforward. What is not at all obvious, however, is how to determine the number of simulation steps which are necessary in order to achieve some specified degree of accuracy. In most application areas, such as Monte Carlo simulations in physics, *ad hoc* arguments are used to estimate this number. There is clearly a need for rigorous analytic bounds if we are to have confidence in the results produced by such methods.

According to our efficiency criterion, the crucial consideration is that the number of steps required for the chain to be close to stationarity should grow only polynomially with the size of the problem description x. We refer to chains which have this property as *rapidly mixing*. Since the number of states will in general be exponentially large, rapid mixing is a strong property: it demands that the process should be close to equilibrium after visiting only a tiny fraction of its state space.

The classical theory of Markov chains is concerned almost exclusively with asymptotic behaviour and provides little useful information about the *rate* of convergence. Recently, several authors have proposed various methods of analysis but these have so far been able to handle only simple chains (such as random walks on cubes) which possess a highly symmetrical structure. The first major contribution of this thesis is to establish a useful characterisation of the rapid mixing property for a wide class of Markov chains. The characterisation is based on a structural property, called the *conductance*, of the weighted graph underlying the chain. The conductance is a global measure of connectedness, or communication strength, of the process and is a natural analogue in this context of the more familiar graph-theoretic concept of *expansion*. Informally, our characterisation says that a (time-reversible) Markov chain is rapidly mixing if and only if the conductance is large. This result represents a generalisation of recent work by Alon and Milman [4], [5] on the connection between the eigenvalues of a graph and its structural properties, whose relevance to the analysis of certain Markov chains has been observed by Aldous [2] and others.

More significant than the characterisation itself is the fact that it allows the rate of convergence of non-trivial Markov chains to be analysed for the first time. In order to do this, we develop a novel general methodology for deriving lower bounds on the conductance of the underlying graphs. Finally, with this machinery in place, we show how to infer the rapid mixing property for natural Markov chains whose states are matchings (independent sets of edges) in a graph. (Chains of this kind were first considered by Broder [12].) We are therefore able to generate almost uniformly in polynomial time the following classes of structures: perfect matchings in a very large class of graphs, including all "dense" graphs, and matchings of all sizes in arbitrary graphs. As a result, we establish for the first time the existence of efficient approximation algorithms for counting these structures. Both counting problems are highly significant, and are

known to be hard in exact form. Counting perfect matchings in bipartite graphs is equivalent to computing the permanent of a 0-1 matrix, a problem for which mathematicians have been seeking an efficient computational procedure for over a century. Counting and generating matchings is of interest in the context of monomer-dimer systems in statistical physics. In the latter case, we are also able to handle natural weighted versions of the problems, and so to approximate the *partition function* of a monomer-dimer system.

We conjecture that the methods described above point the way towards a powerful general approach for analysing the efficiency of algorithms based on finite Markov chains. Potential future application areas include further positive results on approximate counting, rigorous performance guarantees for Monte Carlo simulations in statistical physics, and the demystification of certain stochastic optimisation heuristics such as simulated annealing.

The final theme of the thesis is concerned more with the general notions of approximate counting and almost uniform generation than with particular structures. For approximate counting, we are able to establish a pleasing robustness result: for any self-reducible structures, approximate counting within a factor of the form $1 + n^{\beta}$ is possible in polynomial time either for all real constants β or for no β. Thus we are justified is calling a counting problem *tractable* if it can be aproximated within some such factor by a polynomial time randomised algorithm. This provides a natural way of classifying hard counting problems with respect to a well-defined notion of approximability.

The mechanism used in the above proof is of independent interest. The key step is showing how to generate structures almost uniformly given only very crude counting information for them. This is achieved by means of a Markov chain suggested by the self-reducibility inherent in the problem, which can again be analysed using the characterisation described earlier. This construction can help us to exploit a much wider range of analytic results on approximate counting than has hitherto been possible in this context, even for structures which are not self-reducible.

To illustrate this point, we consider the problem of generating almost uniformly labelled graphs with given vertex degrees and a given excluded subgraph. Using a recent asymptotic result of McKay [40] on the number of such graphs, we are able to generate them in polynomial time from a distribution which is very

close to uniform, provided that the vertex degrees are not too large. The range of degrees we can handle is considerably larger than for previous methods. Moreover, we show the existence of polynomial time algorithms for approximating the number of graphs with a much smaller error than that of available analytic estimates.

The thesis is organised as follows. In Chapter 1 we present some fundamental definitions and summarise earlier work which is relevant to our exposition. The general results on Markov chains are discussed in Chapter 2, culminating in the characterisation of the rapid mixing property. The remaining two chapters are devoted to applications of the characterisation. In Chapter 3 we consider Markov chains for generating specific structures, and establish positive results for the permanent and monomer-dimer systems. Finally, in Chapter 4 we discuss the robustness of our notions of approximate counting and uniform generation, and apply some of our new general machinery to the degree-constrained graph problem mentioned above.

Chapter 1

Preliminaries

The aims of this introductory chapter are twofold: to set down and motivate precise definitions of some fundamental concepts, and to briefly summarise relevant previous knowledge. Accordingly, much of this material is not new, but is included in the interests of orienting the reader.

Three concepts defined here will play a central rôle in all that follows: self-reducibility, approximate counting and almost uniform generation. We trust that the reader will gain an intuitive feel for them through the mostly elementary discussions of this chapter. Almost all the algorithms we shall present in this thesis are probabilistic or randomised. We therefore devote some effort at this stage to a discussion of the model of randomised computation so that technical details may be avoided in the sequel.

The starting point for our later investigations is the work of Jerrum, Valiant and Vazirani [33], in which the intimate computational connection between counting and generation was first formalised. Since this connection underlies most of our later work, we feel it is appropriate to spell it out in some detail in Section 1.4. Finally, we set the scene for the rest of the thesis by identifying a class of combinatorial structures for which the questions of approximate counting and uniform generation are particularly relevant.

1.1 Some basic definitions

We begin by introducing a framework within which a number of natural classes of combinatorial problems, including counting and generation, may be formulated and their computational complexities compared. Common to all classes is the idea of computing certain information about a finite but possibly very large set of combinatorial structures given a concise implicit description of the set. Typically, the description takes the form of some other combinatorial entity drawn from a family of *problem instances*, together with a relation R which associates with each instance x in the family a finite set $R(x)$ of structures called *solutions*, as in the following examples:

1. Problem instances: Boolean formulae B in disjunctive normal form (DNF)
   Solution set $R(B)$: all satisfying assignments of B

2. Problem instances: undirected graphs G
   Solution set $R(G)$: all 1-factors (perfect matchings) of G

3. Problem instances: positive integers n
   Solution set $R(n)$: all partitions of n

More formally, we fix a finite alphabet $\Sigma \supseteq \{0,1\}$ over which instances and solutions are to be encoded, and let $R \subseteq \Sigma^* \times \Sigma^*$ be a binary relation over Σ. For any string $x \in \Sigma^*$, the corresponding solution set with respect to R is just

$$R(x) = \{\, y \in \Sigma^* : \langle x, y \rangle \in R \,\}.$$

Note that no distinction is made between strings which do not encode a "valid" problem instance and those which encode a problem instance with empty solution set. The formal counterpart of example 1 above is then

$$R = \{\, \langle x, y \rangle : x \in \Sigma^* \text{ encodes a Boolean formula } B \text{ in DNF}$$
$$y \in \Sigma^* \text{ encodes a satisfying assignment of } B \,\}.$$

Throughout this thesis we shall move freely between the formal and informal problem descriptions without explicitly saying so, assuming always that the encoding scheme used is "reasonable" in the sense of [24]. The relational framework used here is taken from Jerrum, Valiant and Vazirani [33], and has also

appeared elsewhere, notably in the form of the "search functions" of Valiant [57] and the "string relations" of Garey and Johnson [24].

Our main concern is with classes of problems which involve computing the cardinality of a finite set of combinatorial structures or generating elements of the set uniformly at random. In fact, most of what we will say applies in the more general setting where each structure has an associated positive weight, and the task is to compute the weighted sum of the structures or to generate them randomly with probabilities proportional to their weights. For the sake of simplicity we shall work with unweighted problems until Chapter 3, where weighted structures will arise naturally in one of our applications.

Formally, the *counting problem* for a relation R over Σ involves computing the function $\#R : \Sigma^* \to \mathbb{N}$ defined by $\#R(x) = |R(x)|$. In the (*uniform*) *generation problem* for R, on input $x \in \Sigma^*$, we are asked to select an element of $R(x)$ at random in such a way that each solution has equal *a priori* probability of being chosen. We shall say in the next section precisely what we mean by effective solutions to these problems.

For the purposes of comparison, we mention some other natural problem classes which fit naturally into this framework and which have been extensively studied. Foremost among these is the *existence problem* for a relation R, which on input $x \in \Sigma^*$ asks whether the solution set $R(x)$ is non-empty: in example 1 above, this means asking whether a given DNF formula is satisfiable. The *construction problem* is similar, but requires in addition that some solution $y \in R(x)$ be output if one exists. If some cost function is defined on solutions, then the *optimisation problem* seeks a solution of minimum cost. Finally, the *enumeration problem* for R involves listing all elements of the solution set $R(x)$.

We shall restrict our attention throughout to relations whose solutions are easy to check, i.e., for which membership of a candidate object in a given solution set can be tested efficiently. Let $|x|$ denote the length of the string x. Borrowing terminology from [33], we say that R is a *p-relation* iff

(pr1) There exists a polynomial p such that, for all strings $x, y \in \Sigma^*$,

$$\langle x, y \rangle \in R \Rightarrow |y| \leq p(|x|)\,.$$

(pr2) The predicate $\langle x, y \rangle \in R$ can be tested in time polynomial in $|x| + |y|$.

The class of existence problems associated with p-relations may be identified with the more familiar class NP of languages accepted by polynomially time-bounded nondeterministic Turing machines (NP-machines). Specifically, if M is a nondeterministic Turing machine with at most $|\Sigma|$ possible moves from each configuration, then any halting computation of M may be encoded in a standard way as a string over Σ. It should be clear that R is a p-relation iff there exists an NP-machine M such that, under such an encoding,

$$R = \{ \langle x, y \rangle : y \text{ encodes an accepting computation of } M \text{ on input } x \}.$$

In precisely the same manner, Valiant's class #P [58] may be viewed as the class of counting problems associated with p-relations. (A function $f : \Sigma^* \to \mathbb{N}$ belongs to #P iff there exists an NP-machine M such that, for all $x \in \Sigma^*$, the number of accepting computations of M on input x is $f(x)$.)

For many naturally occurring p-relations, the structures in each solution set have a simple inductive construction from the solution sets of a few smaller instances of the same relation. All of the above examples possess this property: in the case of DNF satisfiability, for instance, there is an obvious (1-1)-correspondence between the satisfying assignments of B and those of the reduced formulae B_T and B_F, which are obtained from B by substituting for one of its variables the values TRUE and FALSE respectively. This property is generally known as *self-reducibility* and was first studied by Schnorr [48]. Formally, we say that a relation R over Σ is (*polynomial time*) *self-reducible* iff

(sr1) There exists a polynomial time computable length function $l_R : \Sigma^* \to \mathbb{N}$ such that $l_R(x) = O(|x|^k)$ for some constant k, and

$$y \in R(x) \Rightarrow |y| = l_R(x) \qquad \forall x, y \in \Sigma^*.$$

(sr2) For all $x \in \Sigma^*$ with $l_R(x) = 0$, the predicate $\Lambda \in R(x)$ can be tested in polynomial time. (Λ denotes the empty string over Σ.)

(sr3) There exist polynomial time computable functions $\psi : \Sigma^* \times \Sigma^* \to \Sigma^*$ and $\sigma : \Sigma^* \to \mathbf{N}$ satisfying

$$\begin{aligned} &\sigma(x) = O(\lg |x|) \\ &l_R(x) > 0 \Leftrightarrow \sigma(x) > 0 && \forall x \in \Sigma^* \\ &|\psi(x,w)| \leq |x| && \forall x, w \in \Sigma^* \\ &l_R(\psi(x,w)) = \max\{l_R(x) - |w|, 0\} && \forall x, w \in \Sigma^* \end{aligned}$$

and such that each solution set can be expressed in the form

$$R(x) = \bigcup_{w \in \Sigma^{\sigma(x)}} \left\{ wy : y \in R(\psi(x,w)) \right\}.$$

Condition (sr3) provides an inductive construction of the solution sets as follows: if the solution length $l_R(x)$ is greater than zero, $R(x)$ can be partitioned into classes according to the (small) initial segment w of length $\sigma(x)$, and each class can then be expressed as the solution set of another instance $\psi(x,w)$, concatenated with w. The partitioning of satisfying assignments of a DNF formula indicated above is easily seen to be of the required form (under some natural encoding). An *atom* is an instance $x \in \Sigma^*$ with solution length $l_R(x) = 0$: in the above example, these would include (encodings of) the constants TRUE and FALSE, viewed as DNF formulae. Condition (sr2) says that, for atoms x, we can test in polynomial time whether $R(x) = \emptyset$ or $R(x) = \{\Lambda\}$. Note that this, together with condition (sr3), implies that we can test in time polynomial in $|x| + |y|$ whether a candidate solution $y \in \Sigma^*$ belongs to the solution set $R(x)$. In view of condition (sr1), any self-reducible relation is therefore automatically a p-relation.

Example 1.1 Consider the relation MATCH which associates with an undirected graph G all matchings (independent sets of edges) of G. Then MATCH is easily seen to be self-reducible since, for an arbitrary edge $e = (u,v)$ of G,

$$\text{MATCH}(G) = \text{MATCH}(G^-) \cup \left\{ M \cup \{e\} : M \in \text{MATCH}(G^+) \right\},$$

where G^- is the graph obtained by deleting the edge e from G, and G^+ the graph obtained by deleting the vertices u and v together with all their incident edges. □

From our point of view, the crucial feature of the above definition is that it allows the counting or uniform generation problem for any instance x to be formulated in terms of the same problem for smaller instances $\psi(x,w)$. We shall exploit this fact later in establishing a close relationship between these two problems for a given self-reducible relation. The following two properties of a self-reducible relation R are easily checked and already well known: if the existence problem for R lies in P, then (i) the construction problem for R can be solved in polynomial time; and (ii) the enumeration problem for R can be solved in time $\#R(x)p(|x|)$, where x is the input and p is a polynomial. In fact, these relationships follow from weaker notions of self-reducibility, in which the strict (1-1)-correspondence of condition (sr3) is replaced by a requirement to the effect that a solution to the existence problem for any instance can be derived easily from solutions for certain subinstances. Such notions have been explored by various other authors (see, e.g., [41], [49]). Despite the fact that our definition appears rather strong, we have to work quite hard to find *natural* relations which cannot easily be formulated so as to be self-reducible. An outstanding candidate is the relation which associates with a positive integer n the set of factors of n.

It is conceptually helpful to capture the inductive construction of solutions of a self-reducible relation explicitly in a tree structure. Suppose that the relation R over Σ is self-reducible. For each $x \in \Sigma^*$ with $R(x) \neq \emptyset$, the *tree of derivations* $T_R(x)$ is a rooted tree in which each vertex v bears both a *problem instance label* $\mathrm{inst}(v) \in \Sigma^*$ and a *partial solution label* $\mathrm{sol}(v) \in \Sigma^*$, defined inductively as follows:

(t1) The root u has labels $\mathrm{inst}(u) = x$ and $\mathrm{sol}(u) = \Lambda$.

(t2) For any vertex v in $T_R(x)$, if the problem instance $z = \mathrm{inst}(v)$ is an atom then v is a leaf. Otherwise, define

$$W(v) = \left\{ w \in \Sigma^{\sigma(z)} : R\left(\psi(z,w)\right) \neq \emptyset \right\}.$$

(Note that $W(v)$ is non-empty.) Then v has a child v_w for each $w \in W(v)$, with labels $\mathrm{inst}(v_w) = \psi(z,w)$ and $\mathrm{sol}(v_w) = \mathrm{sol}(v) \cdot w$.

Note that the labels $\mathrm{sol}(v)$ are distinct, while the $\mathrm{inst}(v)$ are in general not. It should be clear that the labels $\mathrm{sol}(v)$ for leaves v are precisely the elements of $R(x)$, so there is a (1-1)-correspondence between leaves and solutions. More

generally, for any vertex v of $T_R(x)$ there is a (1-1)-correspondence between the solution set $R(\text{inst}(v))$ and the leaves of the maximal subtree rooted at v. The bounds on σ and ψ in the definition of self-reducibility ensure that the depth of $T_R(x)$ is at most $l_R(x)$, and hence bounded by a polynomial in $|x|$, and that its vertex degree is also polynomially bounded.

The definition of $W(v)$ above makes it clear that, in order to infer the structure of the tree of derivations, it is necessary to solve the existence problem for the relation in question. Since we will not always be able to do this with certainty, it is useful to define the *self-reducibility tree* $\tilde{T}_R(x)$ as for $T_R(x)$ above except that the restriction $R(\psi(z,w)) \neq \emptyset$ in the definition of $W(v)$ is removed. Clearly, $\tilde{T}_R(x)$ contains $T_R(x)$ as a subgraph and their labels agree. All solutions in $R(x)$ still occur precisely once as labels of leaves of $\tilde{T}_R(x)$, but there may be other leaves whose partial solution labels are *not* in $R(x)$. The depth and vertex degree of $\tilde{T}_R(x)$ remain polynomially bounded as before.

1.2 Notions of tractability

The aim of this section is to define precisely what is meant by an effective solution to the counting and generation problems introduced earlier. First we must agree on a model of computation: as we will chiefly be studying probabilistic algorithms, some source of randomisation has to be included. It is clearly desirable to formulate definitions in terms of the simplest available model, so we shall use an elementary computing device with access only to a stream of truly random bits. Possible extensions of this model will be considered in the next section.

A *probabilistic Turing machine* (PTM), as defined by Gill [25], is a standard deterministic Turing machine equipped with the additional ability to make decisions according to the fall of an unbiased coin. More formally, a PTM is a Turing machine with output tape [30, Chapter 7] which has certain distinguished "coin-tossing" states. The machine is deterministic except that in coin-tossing states precisely two possible transitions are specified, the choice of transition being determined by the toss of an unbiased coin. If the machine halts, its output is the contents of the output tape; otherwise, its output is undefined. A PTM is *deterministic* if it has no coin-tossing states.

The above definition induces a probability distribution on the set of computations of a PTM M on a given input. We define the random variables $M(x)$ and $\mathrm{Comp}_M(x)$ to denote the output of M on input x and the computation of M on input x respectively, with the proviso that $M(x)$ takes the special value $\perp$ when the output is undefined. We restrict attention to PTMs with alphabet $\Sigma \cup \{?\}$, where $? \notin \Sigma$ is a distinguished symbol which may appear only on the output tape and is to be interpreted as rejection or failure. A computation of a PTM is *accepting* if it halts with some output other than ?. According to the application at hand, various definitions for the time complexity of a PTM are possible. We define the *runtime* of a PTM M on input x to be the length of a longest accepting computation of M on x.

Let us now consider solutions to the counting problem for p-relations. Our approach here is motivated by the observation that efficient *exact* solutions to this problem are the exception rather than the rule. By an efficient solution we mean a deterministic algorithm which computes $\#R(x)$ in time polynomial in the input size $|x|$. Note that the naïve method of explicitly enumerating candidate solutions is unacceptable since in general their number grows exponentially with the size of the input. Indeed, the class #P of counting problems associated with p-relations is not known to lie within any level of the Meyer-Stockmeyer polynomial time hierarchy [53]. More strikingly, there are many natural p-relations (among them examples 1 and 2 of the previous section) whose counting problem is complete for #P, i.e., as hard as any problem in the class, but whose construction problem is solvable in polynomial time: thus while *finding* a perfect matching in a graph is easy, *counting* these structures is apparently intractable. Further examples of this phenomenon can be found in [59].

For functional evaluation problems which have resisted attempts at an exact solution, it is natural to seek efficient approximation algorithms which estimate the value of the function within a specified factor. This notion is familiar from combinatorial optimisation (see, e.g., [24, Chapter 6]) and asymptotic analysis [8]. (A less conventional, and much more severe definition of approximation is studied by Cai and Hemachandra [14].) A second approach towards reducing the complexity of apparently intractable problems is to allow some element of randomisation in the algorithm, requiring only that it delivers a reliable solution with high probability. While rigorous proofs that randomisation helps are avail-

able only in rather restricted circumstances, probabilistic algorithms which run faster than their best known deterministic counterparts have been discovered for a variety of problems (see [60] for a survey), and there has also been considerable recent interest in randomised techniques for combinatorial optimisation [37].

Following Stockmeyer [54] and Karp and Luby [34], we use these ideas to establish a weaker notion of tractability for counting problems. Evidence that apparently hard counting problems may be tractable in this sense is provided by the latter authors, who exhibit a fully-polynomial randomised approximate counter (see below) for satisfying assignments of a DNF formula (example 1 of the previous section).

If a, $\hat{a}$ and r are non-negative real numbers with $r \geq 1$, we say that $\hat{a}$ *approximates* a *within ratio* r if $\hat{a}r^{-1} \leq a \leq \hat{a}r$. Let g be a non-negative integer-valued function on Σ^* and ρ a real-valued function on $\mathbb{N}$ such that $\rho(n) \geq 1$ for all $n \in \mathbb{N}$. A *randomised approximation algorithm for* g *within ratio* ρ is a PTM C which on inputs $x \in \Sigma^*$ always halts with non-negative real-valued output and satisfies

$$\Pr\Big(C(x) \text{ approximates } g(x) \text{ within ratio } \rho(|x|)\Big) \geq 3/4.$$

If C is in fact deterministic, then it is an *approximation algorithm* for g within ratio ρ. In the case that g is the counting function $\#R$ for some relation R over Σ, we call C a (*randomised*) *approximate counter for* R within ratio ρ; if C is deterministic and $\rho(n) = 1$ for all $n \in \mathbb{N}$ then C is an *exact counter* for R. In all cases, C is *polynomially time-bounded* if its runtime is bounded by $p(|x|)$ for some polynomial p and all inputs $x \in \Sigma^*$.

Ideally, we would like to be able to specify the factor in the approximation as part of the input, so that arbitrary accuracy can be achieved (usually at the cost of increased computation time). We say that a PTM C is a *randomised approximation scheme* for g if on inputs $\langle x, \epsilon \rangle \in \Sigma^* \times \mathbb{R}^+$ it always halts with non-negative real-valued output and satisfies

$$\Pr\Big(C(x,\epsilon) \text{ approximates } g(x) \text{ within ratio } 1+\epsilon\Big) \geq 3/4.$$

If $g = \#R$, we speak of a *randomised approximate counter* for R. C is *fully-polynomial* (f.p.) if its runtime is bounded by a polynomial in $|x|$ and ϵ^{-1}. (In practice, we can think of the value ϵ^{-1} as being specified on the input tape as an integer in *unary* notation.)

The significance of the lower bound of 3/4 in the above definitions lies in the fact that it allows the counters to be "powered" so that the probability of producing a bad estimate becomes very small in polynomial time. (This would still hold if 3/4 were replaced by any fixed constant strictly greater than 1/2.) More precisely, we have

Proposition 1.2 *If there exists a polynomially time-bounded randomised approximate counter C for R within ratio ρ, then there exists a PTM C' which on inputs $\langle x, \delta \rangle \in \Sigma^* \times \mathbb{R}^+$ always halts with non-negative real-valued output and satisfies*

$$\Pr\Big(C(x,\delta) \text{ approximates } \#R(x) \text{ within ratio } \rho(|x|) \Big) \geq 1 - \delta,$$

and whose runtime is bounded by a polynomial in $|x|$ and $\lg \delta^{-1}$.

Proof The required procedure C' makes $p(\lg \delta^{-1})$ calls to C, with input x, for a suitable polynomial p and returns the median of the values obtained. For the details, see Lemma 6.1 of [33]. □

Clearly, the powering operation of Proposition 1.2 may be applied in an identical manner to a f.p. randomised approximate counter, which will constitute our primary notion of tractability for counting problems in this thesis.

We now turn our attention to uniform generation problems, taking our notions of tractability from Jerrum, Valiant and Vazirani [33]. It is tempting to demand that an exactly uniform generation algorithm should *always* halt and output some element of the solution set, assuming the latter is non-empty. However, this would not be reasonable within our model since any halting computation of a PTM is performed with probability $1/2^t$ for some $t \in \mathbb{N}$: such a machine would thus not be capable of uniformly generating (say) the elements of a solution set of cardinality 3 in finite time. We therefore allow generators to *fail* (i.e., produce no meaningful output) with bounded probability. Once the possibility of failure is admitted, a number of attractive and natural generation paradigms become available, some of which we shall describe in due course.

As in the case of counting, we also introduce a relaxation of the original problem by allowing the output distribution to deviate from uniformity by a

specified amount, called the *tolerance*. The strict uniformity requirement is seldom of practical importance and if the deviation is small enough it will be effectively undetectable. Furthermore, our notion of "almost uniform" generation arises naturally in a number of ways: as a problem which is polynomial time equivalent to randomised approximate counting for self-reducible relations (Section 1.4); as a generalisation of uniform generation which is robust with respect to extensions of the randomised model (Section 1.3); and from dynamic stochastic algorithms for generation problems (Chapter 2 *et seq*).

Let R be a relation over Σ, and $\epsilon : \mathbb{N} \to \mathbb{R}^+ \cup \{0\}$. A PTM $\mathcal{G}$ is an *almost uniform generator for R within tolerance ϵ* iff its output on input $x \in \Sigma^*$, when defined, belongs to the set $\Sigma^* \cup \{?\}$ and satisfies

(g1) If $R(x) \neq \emptyset$ then $\Pr\big(\mathcal{G}(x) \in \Sigma^*\big) \geq 1/2$

(g2) There exists a function $\phi : \Sigma^* \to (0,1]$ such that, for all $y \in \Sigma^*$,

$$y \notin R(x) \Rightarrow \Pr\big(\mathcal{G}(x) = y\big) = 0$$
$$y \in R(x) \Rightarrow \big(1+\epsilon(|x|)\big)^{-1}\phi(x) \leq \Pr\big(\mathcal{G}(x) = y\big) \leq \big(1+\epsilon(|x|)\big)\phi(x).$$

Note that the generator may only output valid solutions or the failure symbol ?, and that it either fails or runs forever when the solution set is empty. $\mathcal{G}$ is a *uniform generator* for R if $\epsilon(n) = 0$ for all $n \in \mathbb{N}$. In either case, $\mathcal{G}$ is *polynomially time-bounded* if its runtime is bounded by a polynomial in $|x|$.[1] If $\mathcal{G}$ is polynomially time-bounded, we may attach a clock to it and force all non-accepting computations to halt within the time bound with output ?. We will assume that all such generators have this property.

By analogy with counters, it should be clear how to define an *almost uniform generator* $\mathcal{G}$ for R, whose tolerance $\epsilon > 0$ can be preset as part of the input. Specifically, (g2) above now becomes

---

[1] In many contexts, an appropriate complexity measure for a PTM is its *expected* runtime. In the case of generation, however, a bound on the expected runtime leaves open the undesirable possibility that certain solutions only ever appear after a very long time. For approximate counters, either measure will do.

(g2′) There exists a function $\phi : \Sigma^* \times \mathbb{R}^+ \to (0,1]$ such that, for all $y \in \Sigma^*$,

$$y \notin R(x) \Rightarrow \Pr\big(\mathcal{G}(x,\epsilon) = y\big) = 0$$
$$y \in R(x) \Rightarrow (1+\epsilon)^{-1}\phi(x,\epsilon) \leq \Pr\big(\mathcal{G}(x,\epsilon) = y\big) \leq (1+\epsilon)\phi(x,\epsilon),$$

where the input $\langle x, \epsilon\rangle \in \Sigma^* \times \mathbb{R}^+$. $\mathcal{G}$ is then *fully-polynomial* if its runtime is bounded by a polynomial in $|x|$ and $\lg \epsilon^{-1}$. Note that the definition of f.p. here differs from that for approximate counters in the presence of the logarithm: the errors tolerated in generation are exponentially small. (The value ϵ^{-1} may be thought of as being specified on the input tape as an integer in *binary* notation.) As before, f.p. generators are assumed always to halt within their time bound.

Clearly, if $R(x) \neq \emptyset$ the probability of failure of any of the time-bounded generators defined above may be reduced rapidly by repeated trials. More precisely, $\lg \delta^{-1}$ iterations of the generator suffice to ensure that the failure probability does not exceed δ. By the same token, we could replace $1/2$ in the definition by $1/p(|x|)$ for an arbitrary polynomial p. Furthermore, if the construction problem for a relation R is solvable deterministically in polynomial time then a f.p. almost uniform generator for R may be modified so as *never* to fail when $R(x)$ is non-empty.

In this thesis, we shall regard the existence of a f.p. generator as an effective criterion for tractability of generation problems. We should point out that the gap between exact and approximate notions here is apparently considerably smaller than in the case of counting. For example, viewed as black boxes, almost uniform and uniform generators will be effectively indistinguishable under any reasonable definition of a statistical test involving polynomially many observations. Moreover, we know of no structures which can be generated almost uniformly in an efficient manner but for which exactly uniform generation is (in some appropriate sense) hard. As we shall see in Section 1.5, such a dichotomy does arise in the case of approximate counting.

For future reference, we also note here a standard randomised definition of tractability for existence problems. The existence problem for a relation $R \subseteq \Sigma^* \times \Sigma^*$ belongs to the class RP ("randomised polynomial time") iff there exists a PTM M with polynomially bounded runtime which behaves as follows:

for all inputs $x \in \Sigma^*$,

$$R(x) \neq \emptyset \Rightarrow \Pr(M \text{ accepts } x) \geq 1/2$$
$$R(x) = \emptyset \Rightarrow \Pr(M \text{ accepts } x) = 0.$$

The probability of M making an error is therefore one-sided, and can be made to decay very rapidly by the standard device of repeated trials. The class RP (also called VPP by Gill [25]) replaces P as the class of tractable decision problems for randomised computation. Note that $\mathrm{P} \subseteq \mathrm{RP} \subseteq \mathrm{NP}$, and, since problems in RP are intuitively tractable, the second inclusion is widely conjectured to be strict. The first is also thought to be strict for different reasons.

1.3 An extended model

Using j successive tosses of its fair coin, a PTM can clearly simulate just those branching probabilities which are of the form $i2^{-j}$, for $i \in \{0, 1, \ldots, 2^j\}$. The inability of a PTM to branch with more general probabilities often makes the design and justification of algorithms needlessly complicated. Indeed, randomised algorithms in the literature are typically expressed in terms of more general branching probabilities involving the ratio of two previously computed integers. Accordingly, in this section we introduce an extension of the PTM model which can easily simulate this kind of branching behaviour. We go on to establish sufficient conditions for counting and generation algorithms in this model to be efficiently implementable by a PTM. This will allow us in the sequel to work with the extended model while retaining our basic definitions of tractability.

An *oracle coin machine* (OCM) is a generalisation of a PTM in which the unbiased coin is replaced by a coin whose bias can be set and varied by the machine itself. The bias of the coin is determined by the contents of a special bias tape: if in a coin-tossing state this tape contains the encoding of a rational number $q = r/s$, where r, s are coprime natural numbers in binary with $r \leq s$ and $s > 0$, then the two possible transitions are chosen with probabilities q and $1 - q$ respectively; otherwise, the machine halts immediately with distinguished output ?. In all other respects, an OCM is identical to a PTM. The reason for rejecting such a machine as the primary randomised model is that an elementary branching step could not reasonably be implemented by a physical device in constant time.

As is customary, Turing machine algorithms will be expressed in a dialect of Pidgin Algol. The OCM model allows this language to be augmented with the probabilistic branching commands defined below, whose semantics should be obvious. In what follows, q, $\{q_i\}$ denote rationals in the interval $[0,1]$, n, m natural numbers, and P, $\{P_i\}$ programs.

1. (Two-way branching)

   either (with prob q**)** P_0
   or (with prob $1-q$**)** P_1

2. (Probabilistic execution)

   with prob q **do** P

3. (m-way branching)

   either (with prob q_0**)** P_0
   or (with prob q_1**)** P_1
   $\vdots \qquad \vdots$
   or (with prob q_{m-1}**)** P_{m-1}

   where $m > 1$ and $\sum_i q_i = 1$.

4. (Weighted selection from an m-set)

   select $j \in \{0,1,\ldots,m-1\}$ **with prob** $q_j / \sum_i q_i$

   where $m > 0$. (If $\sum_i q_i = 0$, the machine should halt with output ?.)

5. (Uniform selection form an ordered n-set)

   select $j \in \{0,1,\ldots,n-1\}$ **u.a.r.**

   where $n > 0$.

Note that the integer elements in 4 and 5 may be viewed as indices, thus allowing selection from general ordered sets.

Proposition 1.3 *Each of the above branching operations can be performed by an OCM in runtime bounded by a polynomial in m and the sizes of the binary representations of q, $\{q_i\}$ and n, plus the maximum runtime of the programs P, $\{P_i\}$.*

Proof

1. This is the basic OCM branching operation. The time required, in addition to that for P_0 and P_1, is a single branching step plus the time to write the encoding of q on the bias tape.

2. A special case of 1 in which P_1 is null.

3. By performing appropriate tests and relabelling if necessary, we may assume w.l.o.g. that all the q_i are greater than 0. The OCM first computes the quantities $q_i' = q_i/(1 - \sum_{j=0}^{i-1} q_j)$ for $0 < i < m-1$, and $q_0' = q_0$. It then executes the following sequence of two-way branchings:

   either (**with prob** q_0') P_0
   or (**with prob** $1 - q_0'$)
   either (**with prob** q_1') P_1
   or (**with prob** $1-q_1'$)
   $\vdots$ $\vdots$
   either (**with prob** q_{m-2}') P_{m-2}
   or (**with prob** $1 - q_{m-2}'$) P_{m-1}

   Clearly, both the arithmetic and the branching can be performed in time bounded by a polynomial in m and the sizes of the representations of $\{q_i\}$.

4. This is essentially equivalent to 3.

5. Consider the following recursive procedure which uses only two-way branchings:

   procedure *select*(a, b);
   begin
   $l := b - a + 1$;
   if $l = 1$ **then** $j := a$

```
    else begin
        m := (a + b) div 2;  l' := m − a + 1;
        either (with prob l'/l) select(a, m)
           or (with prob 1 − l'/l) select(m + 1, b)
    end
end
```

The call $select(0, n-1)$ achieves the desired effect and leads to at most $O(\log n)$ levels of recursion, while the arithmetic is trivial. The total runtime is therefore polynomial in the size of the representation of n. □

The next issue we have to address is the effective implementation of OCM algorithms in the more restricted PTM model. We shall do this with reference to the notions of tractability for counting and generation problems introduced earlier. (Analogous questions for existence problems with respect to a similar extended model have been studied by Lautemann [38].) In the following, we refer to OCMs which serve as counters or generators for a relation R over Σ in the various senses already introduced: the definitions differ from those of the previous section only in the model of computation.

First let us observe that, as far as our most important approximate notions of counting and generation are concerned, the class of tractable problems is *robust* with respect to the change of model.

Proposition 1.4 *If there exists a f.p. OCM randomised almost uniform generator (respectively, a f.p. OCM randomised approximate counter) for R, then there exists a f.p. almost uniform generator (respectively, a f.p. randomised approximate counter) for R.*

Proof Let $\mathcal{G}$ be the postulated OCM generator for R, and let p be a polynomial bounding its runtime. We will describe a PTM $\mathcal{G}'$ which on input $\langle x, \epsilon \rangle \in \Sigma^* \times \mathbb{R}^+$ approximately simulates the computation process of $\mathcal{G}$ on input $\langle x, \epsilon/3 \rangle$.

Assume without loss of generality that $\epsilon \leq 1$, and set $m = \max\{p(|x|, \lg(3/\epsilon)), 4\}$: then m bounds the runtime of $\mathcal{G}$ on input $\langle x, \epsilon/3 \rangle$, and all non-trivial branching probabilities of $\mathcal{G}$ must lie in the range $[2^{-m}, 1 - 2^{-m}]$. The idea is that, for any such branching probability q, $\mathcal{G}'$ will compute a good approximation to q

of the form $i_q 2^{-j}$, where $i_q \in \mathbb{N}$ and $j = 2m + \lg(3/\epsilon) + 1$. Specifically, if we set $i_q = \lfloor q\, 2^j \rfloor$, then the rationals $i_q 2^{-j}$ and $1 - i_q 2^{-j}$ approximate q and $1 - q$ respectively within ratio $1 + 2^{-m}\epsilon/3$, as may easily be verified.

The simulation proceeds as follows. Deterministic steps of $\mathcal{G}$ are simulated directly by $\mathcal{G}'$, and if $\mathcal{G}$ halts then $\mathcal{G}'$ halts, necessarily with the same output. To simulate a non-trivial branching step of $\mathcal{G}$ of the form

$$\begin{aligned} &\textbf{either (with prob } q\textbf{) } P_0 \\ &\quad \textbf{or (with prob } 1-q\textbf{) } P_1 \end{aligned} \tag{1.1}$$

$\mathcal{G}$ tosses its fair coin j times and performs the branching

$$\begin{aligned} &\textbf{either (with prob } i_q/2^j\textbf{) simulate } P_0 \\ &\quad \textbf{or (with prob } 1 - i_q/2^j\textbf{) simulate } P_1 \end{aligned}$$

The time required to compute i_q and to do the coin tossing is clearly bounded by a polynomial in m and $\lg \epsilon^{-1}$. Since there are at most m steps of $\mathcal{G}$ to simulate, the runtime of $\mathcal{G}'$ is bounded by a polynomial in $|x|$ and $\lg \epsilon^{-1}$, as required.

Now suppose that $\mathcal{G}$ performs some computation c with probability $p(c)$. By our earlier remarks, the probability that $\mathcal{G}'$ simulates c, yielding the same output, approximates $p(c)$ within ratio

$$(1 + 2^{-m}\epsilon/3)^m \leq 1 + m^2 2^{-m}\epsilon/3 \leq 1 + \epsilon/3\,,$$

where we have used the binomial theorem and the facts that $m \geq 4$ and $\epsilon \leq 1$. Thus the tolerance in the output distribution of $\mathcal{G}'$ is at most $(1 + \epsilon/3)^2 \leq 1 + \epsilon$. Applying the same argument to non-accepting computations, we see that the failure probability of $\mathcal{G}'$ exceeds that of $\mathcal{G}$ by at most a factor of $1 + \epsilon/3 \leq 4/3$, and so is bounded above by $2/3$. Thus $\mathcal{G}'$, modified to allow a single repeated trial on failure, is a f.p. almost uniform generator for R.

The proof for the counter is identical, except that the final modification makes use of the powering operation of Proposition 1.2. □

In very similar fashion, we can show the following:

Proposition 1.5 *If there exists a polynomially time-bounded OCM almost uniform generator for R within tolerance $\epsilon(n)$ (respectively, a polynomially time-bounded OCM randomised approximate counter for R within ratio $\rho(n)$), then*

there exists a polynomially time-bounded almost uniform generator for R within tolerance $\epsilon(n) + 2^{-p(n)}$ for any polynomial p (respectively, a polynomially time-bounded randomised approximate counter for R within ratio $\rho(n)$). □

Exact generators differ from the above devices in that their output distribution is prescribed precisely up to the probability of failure, so the straightforward approach of the previous proof is not enough here. Before dealing with this problem, it is useful to formalise the concept of simulation in the present context.

Let M be an OCM and $T(x)$ its runtime on input x. A PTM M' is said to *efficiently simulate* M iff

(s1) For all inputs x for M and all strings $y \neq ?$,

$$\Pr\big(M'(x) = y\big) = \psi(x) \Pr\big(M(x) = y\big),$$

where $\psi(x) \geq 1/p_1(|x|)$ for some polynomial p_1.

(s2) For all inputs x for M, the runtime of M' on x is at most $p_2(|x|)T(x)$ for some polynomial p_2.

(Note that the approximate simulation in the proof of Proposition 1.4 could also be formalised in this way, though $\psi(x)$ would have to be replaced by a small range of values.) Of course, this is a minimal requirement since it demands only that the output distribution of M (conditional on acceptance) be preserved. The simulation is efficient in the sense that the runtime of M' is not much greater than that of M, and that if the failure probability of M is bounded away from 1 then that of M' may be similarly bounded using only a polynomial number of repeated trials. The following observation is immediate from the above definition:

Proposition 1.6 *Suppose that M is a polynomially time-bounded OCM uniform generator for R, and that the PTM M' efficiently simulates M. Then M', suitably powered to reduce its failure probability, is a polynomially time-bounded uniform generator for R.* □

Next we derive a simple sufficient condition for efficient simulation by a PTM to be possible. The condition is expressed in terms of the probability distribution over the accepting computations of the OCM.

Theorem 1.7 *Let M be an OCM with polynomially bounded runtime. Suppose there exists a polynomial time computable function $N : \Sigma^* \to \mathbb{N}^+$ with the property that, for all inputs $x \in \Sigma^*$ and all accepting computations c of M on input x, the quantity*

$$N(x)\Pr\big(\mathrm{Comp}_M(x) = c\big)$$

is integral. Then M can be efficiently simulated by a PTM.

Proof Let p be a polynomial bounding the runtime of M: since we are only required to simulate accepting computations, we may assume that all non-accepting computations of M halt within this time bound with output ?. Let x be an input for M, and set $m = p(|x|)$. Clearly, all non-trivial branching probabilities of M on this input must lie in the range $[2^{-m}, 1 - 2^{-m}]$.

The required PTM M' operates in two consecutive phases: a simulation phase and a correction phase. The first of these is very similar to the simulation in the proof of Proposition 1.4, except that M' must be careful to select its branching probabilities from a small set so that subsequent correction is possible. Specifically, M' works with the set $A = \{\lceil 2^{i/2m}\rceil 2^{-2m} : 1 \le i \le 4m^2\}$ of cardinality $4m^2$. It is not hard to see [33, Lemma 2] that for each real $q \in [2^{-m}, 1 - 2^{-m}]$ there exists $a_q \in A$ such that

$$a_q \le q \le (1 + 1/m)a_q\,, \tag{1.2}$$

so that in particular a_q approximates q within ratio $1 + 1/m$.

As before, M' simulates deterministic steps of M directly. However, each non-trivial branching of the form (1.1) is now simulated by M' using the branching

either (with prob a_q) simulate P_0
or (with prob a_{1-q}) simulate P_1
or (with prob $1 - a_q - a_{1-q}$) halt with output ?

From the definition of A, this can be realised by M' using only $2m$ tosses of its fair coin. Assuming no failure, the simulation phase continues until M reaches a halting configuration. If the corresponding output of M is ? then M' also halts with output ?; otherwise, M' enters its correction phase, to be described shortly. The total time required for the simulation phase is clearly bounded by a polynomial in $|x|$.

Upon entry to the correction phase, M' has simulated some accepting computation c of M. By maintaining two running products, we may assume that M' has computed both the probability $p(c) = \Pr(\mathrm{Comp}_M(x) = c)$ (i.e., the product of the branching probabilities it has simulated), and the probability $p'(c) = \Pr(\mathrm{Comp}_{M'}(x)$ simulates $c)$ (i.e., the product of the branching probabilities actually used). By (1.2), each term in the second product approximates the corresponding term in the first within ratio $1 + 1/m$, and we have

$$p'(c) \leq p(c) \leq (1 + 1/m)^m p'(c) \leq \mathrm{e}\, p'(c)\,. \tag{1.3}$$

Furthermore, M' is able to compute in polynomial time a natural number $N(x)$ such that $N(x)p(c)$ is integral, as stipulated in the statement of the theorem.

Denote by $\pi(A)$ the product $\prod_{a \in A} a^m$. The correction phase of M' now consists of a single branching of the form

$q := \pi(A)N(x)2^k p(c)/p'(c)$;
either (with prob q**) halt** with output y
or (with prob $1 - q$**) halt** with output ?

where y is the output of the simulated computation c and k is an integer independent of c to be specified below: in particular, k will be chosen so that $q \leq 1$. It should be clear from the form of q that the above branching can be achieved by M' using its fair coin. Furthermore, the probability that M' simulates c and, after correction, accepts with the same output is

$$\left(\pi(A)N(x)2^k\right) p(c) = \psi(x)p(c)\,,$$

where $\psi(x)$ is independent of c. For the simulation to be efficient, we require a lower bound on ψ, which we get by choosing k appropriately. Recall that the only constraint on the choice of k is that $q \leq 1$. But from (1.3) we have

$$q = \frac{p(c)}{p'(c)}\psi(x) \leq \mathrm{e}\,\psi(x)\,,$$

so we may select k to be maximal such that $\psi(x) = \pi(A)N(x)2^k \leq 1/\mathrm{e}$. It then follows that $\psi(x) > 1/2\mathrm{e}$ is bounded below as required.

The above discussion indicates that M' satisfies condition (s1) of the definition. For condition (s2) we need only make the additional observation that the arithmetic of the correction phase can be performed in polynomial time since the cardinality of A is only $4m^2$. Hence the runtime of M' is polynomially bounded as required. □

The sufficient condition of Theorem 1.7 may at first sight appear cumbersome to apply in practice. However, most known generation algorithms can readily be seen to satisfy it. One typical case is an OCM uniform generator each of whose accepting computations yields a distinct solution as output. For a fixed input x, any such computation must therefore be executed with probability r_x/s_x for constants r_x, s_x which are known *a posteriori*, and we may take $N(x) = s_x$. A slightly less trivial example is provided by the following algorithm, due to Nijenhuis and Wilf [44], for uniformly generating partitions of a positive integer.

Example 1.8 A *partition* of a positive integer n is a representation π of the form

$$n = r_1 + \cdots + r_l$$

for integers r_i with $r_1 \geq r_2 \geq \cdots \geq r_l > 0$. We shall identify partitions of n with multisets of the form $\{\mu_1 \cdot r_1, \ldots, \mu_k \cdot r_k\}$, where $\mu_i, r_i \in \mathbb{N}^+$, the r_i are distinct and $\sum_{i=1}^{k} \mu_i r_i = n$. For convenience, n is allowed to be zero, in which case the only partition is the empty set.

Define the relation Π which associates with each $n \in \mathbb{N}$ the set of partitions of n, and consider the recursive procedure *Genpart* of Figure 1.1. Here $\pi + j \cdot r$ denotes the operation of adjoining j copies of r to the multiset π. It is assumed in line (2) that the counting function $\#\Pi$ for Π is computed by some other procedure (using, e.g., the recurrence relation implicit in *Genpart* itself), and that $\#\Pi(m) = 0$ for $m < 0$.

We claim that the call *Genpart*$(n, \varnothing)$ uniformly generates partitions of n. More generally, we show by induction on n that the call *Genpart*(n, π) uniformly generates partitions of n and adjoins them to π. The base case $n = 0$ is immediate from line (1). Suppose then that $n > 0$ and consider the partition π' of n given by

$$\pi' = \{\mu_1 \cdot r_1, \ldots, \mu_k \cdot r_k\}.$$

Examination of the algorithm reveals that π' may be adjoined to π in precisely $\mu_1 + \cdots + \mu_k$ ways as follows: in line (3) select $r = r_i$ for some $1 \leq i \leq k$, and j in the range $1 \leq j \leq \mu_i$; finally, adjoin to $\pi + j \cdot r_i$ the partition

$$\{\mu_1 \cdot r_1, \ldots, (\mu_i - j) \cdot r_i, \ldots, \mu_k \cdot r_k\}$$

of $n - jr_i$ via the recursive call in line (4). By the inductive hypothesis, the recursive call generates partitions uniformly. In view of lines (2) and (3), the

```
procedure Genpart(n : natural_number; π : partition);
begin
(1)   if n = 0 then halt with output π
      else begin
(2)       for ⟨r,j⟩ ∈ [1,n] × [1,n] do q(r,j) := r#Π(n − jr);
(3)       select ⟨r,j⟩ ∈ [1,n] × [1,n] with prob q(r,j)/Σq(r',j');
(4)       Genpart(n − jr , π + j · r)
      end
end
```

Figure 1.1: Procedure for uniformly generating partitions

probability that π' is appended to π is therefore

$$\sum_{i=1}^{k}\sum_{j=1}^{\mu_i} \frac{r_i \#\Pi(n-jr_i)}{Q(n)\,\#\Pi(n-jr_i)} = \frac{1}{Q(n)}\sum_{i=1}^{k}\mu_i r_i = \frac{n}{Q(n)},$$

where $Q(n) = \sum q(r',j')$ as in line (3) and depends only on n. Since π' was chosen arbitrarily, we conclude that the generation is uniform and also that $Q(n) = n\#\Pi(n)$. For a combinatorial derivation of this procedure, the reader is referred to [44].

It should be clear that *Genpart* can be implemented by a polynomially time-bounded OCM. (Note that the evaluation of $\#\Pi$ at the appropriate points can be performed efficiently.) Now let c be any computation of such a machine on input $\langle n,\varnothing\rangle$ with $n > 0$. The form of the selection probabilities in line (3), together with the fact that $Q(m) = m\#\Pi(m)$ for all m, implies that c is executed with probability

$$p(c) = \frac{t_1\#\Pi(m_2)}{m_1\#\Pi(m_1)}\,\frac{t_2\#\Pi(m_3)}{m_2\#\Pi(m_2)}\cdots\frac{t_l\#\Pi(0)}{m_l\#\Pi(m_l)} = \frac{1}{\#\Pi(n)}\prod_{i=1}^{l}\frac{t_i}{m_i}$$

for some $t_i, m_i, l \in \mathbb{N}^+$, with $n = m_1 > m_2 > \cdots > m_l > 0$. Hence the product $n!\,\#\Pi(n)\,p(c)$ is always integral. We may therefore take $N(n) = n!\,\#\Pi(n)$ in Theorem 1.7 and deduce that the algorithm has a PTM implementation. □

To conclude this discussion of models, we further motivate Theorem 1.7 by showing that efficient simulation by a PTM is not possible for arbitrary polynomially time-bounded OCMs. In other words, the extended model is strictly more

powerful than the PTM model in terms of its ability to realise a given output distribution exactly in polynomial time.

Theorem 1.9 *There exists a polynomially time-bounded OCM which cannot be efficiently simulated by any PTM.*

Proof Consider the OCM M which, on any input x of length n, executes the following:

> **select** $j \in \{1, \ldots, 2^n\}$ **u.a.r.**;
> **either (with prob** $1/j$**) halt** with output ?
> **or (with prob** $1 - 1/j$**) halt** with output j

By Proposition 1.3, the runtime of M is bounded by a polynomial in n. Furthermore, for each j in the range $1 \leq j \leq 2^n$, M outputs j with probability $(j-1)(2^n j)^{-1}$. Note incidentally that M is *effective* in the sense that its failure probability on inputs of length n is $1 - H_{2^n}/2^n$, which tends to zero exponentially fast with n. (Here H_k denotes the kth harmonic number.)

Now suppose there exists a PTM M' which efficiently simulates M. Since M is polynomially time-bounded, M' must be also, by condition (s2). So let p be a polynomial bounding its runtime, and set $m = p(n)$. Then clearly, for any input x of length n and integer j, we must have

$$\Pr\bigl(M'(x) = j\bigr) = i(j)2^{-m}$$

for some integer $i(j)$ in the range $[0, 2^m]$.

Given the output distribution of M, condition (s1) of the definition therefore demands that

$$i(j) = \eta(x)\,\frac{j-1}{j} \tag{1.4}$$

for $1 \leq j \leq 2^n$ and some constant $\eta(x) > 0$ independent of j. Setting $j = 2$ in (1.4) implies that $\eta(x) = 2i(2)$, so $\eta(x)$ is integral and bounded above by 2^{m+1}. Furthermore, since $i(j)$ itself is integral, (1.4) also implies that $j \mid \eta(x)$ for $1 \leq j \leq 2^n$. However, it is well known that, for infinitely many natural numbers r, $\mathrm{lcm}\{1, \ldots, r\} \geq Ae^r$ for some constant A (see, e.g., [22, page 34]). We must therefore have

$$\eta(x) \geq \mathrm{lcm}\{1, \ldots, 2^n\} \geq k\,\mathrm{e}^{2^n}$$

infinitely often, which is a contradiction since $\eta(x)$ is bounded above by $2^{m+1} = 2^{p(n)+1}$, and p is a polynomial. We must therefore conclude that the integers $i(j)$ do not always exist as claimed, so M' does not efficiently simulate M. □

It could be argued that the above counterexample exploits the special properties of a bizarre output distribution, and that a similar result would not necessarily hold if attention were restricted to (say) uniform generators. However, we claim that our weak definition of simulation makes it entirely reasonable to consider general distributions. This is because we are really only using outputs to encode distinct *computations* of the OCM. According to our definition, the algorithm of the simulating machine need bear no resemblance to that of the original OCM. A more realistic definition of simulation would require that each individual computation step be simulated explicitly. (Note that the simulations of Proposition 1.4 and Theorem 1.7 have this property.) Under this definition, the proof of Theorem 1.9 still holds even when the OCM M is modified so as to have a trivial output distribution.

The question of whether there exists a relation which can be *uniformly* generated in polynomial time by an OCM but by no PTM is open, and seems difficult. In view of Theorems 1.7 and 1.9, this is not inconceivable but any efficient OCM generator for such a relation must necessarily induce quite a complex distribution on its set of accepting computations.

1.4 Counting, generation and self-reducibility

In this section we describe two reductions due to Jerrum, Valiant and Vazirani which together establish a close relationship between the complexities of approximate counting and almost uniform generation for most interesting p-relations. In each case, the additional property which is needed to make the reduction work is self-reducibility. Similar results in a less general setting have been proved by Broder [12].

Most known uniform generation algorithms for combinatorial structures may be viewed as instances of the following generic reduction to the corresponding counting problem. Suppose that the structures have a simple inductive construction, or more formally that they may be described in terms of a self-reducible

```
      procedure GenR(v : vertex);
      begin
(1)       z := inst(v);
(2)       if l_R(z) = 0 then halt with output sol(v)
          else begin
(3)           select w ∈ Σ^σ(z) with prob C(ψ(z,w))/Σ_w' C(ψ(z,w'));
(4)           inst(u) := ψ(z,w);   sol(u) := sol(v) · w;
(5)           GenR(u)
          end
      end
```

Figure 1.2: Reduction from generation to counting

relation R, and let x be a problem instance with non-empty solution set. Now select a random path from the root of the tree of derivations $T_R(x)$ to a leaf (solution), choosing the next edge at each stage with probability proportional to the number of solutions in the maximal subtree rooted at its lower end: this information may be obtained by evaluating the function $\#R$ for appropriate problem instance labels in the tree. It is then easy to see that the distribution over solutions is uniform.

A formal specification of the algorithm is provided by the recursive procedure *GenR* shown in Figure 1.2, in which σ, ψ, **sol**, **inst** have the meanings ascribed to them in the earlier definitions and C is an exact counter for R, viewed as an oracle. Elements of $R(x)$ are generated using the call $GenR(u)$, where u represents the root of $T_R(x)$, i.e., $\mathrm{inst}(u) = x$ and $\mathrm{sol}(u) = \Lambda$. Since the depth and vertex degree of the tree are polynomially bounded, the algorithm can be implemented on a polynomially time-bounded OCM equipped with the oracle C for $\#R$. Furthermore, as each computation yields a distinct output, Theorem 1.7 ensures that a PTM implementation exists.

The above procedure can actually still be made to work when the counting information supplied by the oracle C is slightly inaccurate, specifically if it is within ratio $1 + \mathrm{O}(n^{-k_R})$, where $k_R > 0$ is a constant satisfying $l_R(x) = \mathrm{O}(|x|^{k_R})$ (recall condition (sr1) in the definition of self-reducibility). This is done by appending a correction step prior to the output in line (2), much as in the

proof of Theorem 1.7. To see how this works, view the uncorrected procedure as an approximate simulation of the ideal procedure with exact oracle: given the constraint on the counting estimates, each branching probability in line (3) is simulated within ratio $\left(1 + c/l_R(\psi(x,w))\right)^2$ for some fixed constant $c \geq 1$. Taking the product of such factors along any path from root to leaf, and bearing in mind that the length function l_R decreases strictly along the path, we find that the generation probability $p(y)$ of each solution $y \in R(x)$ approximates the ideal value $1/\#R(x)$ within ratio

$$\prod_{i=1}^{l_R(x)-1} \left(1 + \frac{c}{i}\right)^2 \leq \left(\prod_i \left(1 + \frac{1}{i}\right)\right)^{2c} = l_R(x)^{2c}. \tag{1.5}$$

Writing $m = l_R(x)$, this implies that $p(y)$ is bounded below by

$$p(y) \geq \frac{m^{-2c}}{\#R(x)} \geq \frac{m^{-2c}}{(1 + c/m)C(x)} = \phi(x). \tag{1.6}$$

The correction step is now simply a matter of outputting y with probability $\phi(x)/p(y)$, which by (1.6) is ≤ 1, and failing otherwise. Each solution is thus generated with the uniform probability $\phi(x)$, and the probability that no failure occurs is

$$\#R(x)\,\phi(x) \geq \frac{m^{-2c}}{(1 + c/m)^2}, \tag{1.7}$$

which may be boosted to $1/2$ as usual using only polynomially many repeated trials. Implementation in the PTM model follows as before.

Finally, consider what happens if we use the same procedure when C is a *randomised* approximate counter within the above ratio. Assuming initially that *all* values returned by C are within this ratio, the previous argument still holds and solutions are generated with some uniform probability $\phi(x)$ which is again bounded as in (1.7). Unfortunately, we can say nothing about the output distribution when some value returned by C happens to be very inaccurate: however, by appealing to Proposition 1.2, we may ensure that C behaves badly with such small probability δ that the resulting procedure (with C as oracle) is a f.p. almost uniform generator for R.

To see how to choose δ, note first that the total number of oracle calls is bounded by $q(|x|)$, where q is a polynomial which depends on the depth and vertex degree of the tree. The probability that *any* returned value falls outside the acceptable range is therefore at most $\delta q(|x|)$, and the effect of this event on

the output probability $p(y)$ of any solution y is thus an additive term $\pm\delta q(|x|)$, i.e.,

$$\phi(x) - \delta q(|x|) \leq p(y) \leq \phi(x) + \delta q(|x|)\,. \tag{1.8}$$

(Note also that, since the oracle no longer decides with certainty whether a given solution set is non-empty, the process is not now confined to the tree of derivations but may occasionally fall into other parts of the self-reducibility tree. As a result, some non-solution leaves may be reached with non-zero probability: however, since R is a p-relation the condition $\langle x, y\rangle \in R$ may always be checked prior to output.) Now observe that, since solutions are strings of length m over Σ, their total number $\#R(x)$ cannot exceed $|\Sigma|^m$. Combining this with the bound (1.7) on $\phi(x)$, we see that by choosing

$$\delta = \frac{(\epsilon/2)m^{-2c}}{(1+c/m)^2|\Sigma|^m q(|x|)} \leq \frac{(\epsilon/2)\phi(x)}{q(|x|)}\,,$$

we can write the additive approximation (1.8) of $\phi(x)$ by $p(y)$ as a *relative* approximation within ratio $1+\epsilon$ (assuming $0 < \epsilon \leq 1$), so that the resulting generator is almost uniform within tolerance ϵ. Since the powering operation requires time polynomial in $\lg \delta^{-1}$, the runtime of the procedure with C as oracle is polynomial in $\lg \epsilon^{-1}$ and $|x|$, as required. Implementation on a PTM follows directly from Proposition 1.4. If C itself is a polynomially time-bounded counter, then the entire procedure constitutes a f.p. almost uniform generator for R.

The above discussion is summarised in the following theorem:

Theorem 1.10 (Jerrum, Valiant and Vazirani) *Let R be a self-reducible relation over Σ and k_R a constant such that, for all pairs $\langle x, y\rangle \in R$, $|y| = \mathrm{O}(|x|^{k_R})$. If there exists a polynomially time-bounded randomised approximate counter for R within ratio $1+\mathrm{O}(n^{-k_R})$ then there exists a f.p. almost uniform generator for R. Moreover, if the counter is deterministic then there exists a polynomially time-bounded uniform generator for R.* □

Note that the ratio $1+\mathrm{O}(n^{-k_R})$ appears to be a threshold value for this simple reduction technique: under a weaker hypothesis, there seems to be no polynomial bound on the product (1.5), indicating that the cumulative errors may then become too large for effective correction to be possible.

As we have already mentioned, numerous generation algorithms may readily be derived from the above reduction to *exact* counting. Several simple examples

appear in the book by Nijenhuis and Wilf [44], who also present an alternative formulation of the generic reduction in this case. (Note that the algorithm of Example 1.8 can be recast in this form by modifying the underlying relation and biasing the generation probabilities appropriately.) Other examples in the literature include generating spanning trees in a graph [26], terminal strings of given length in an unambiguous context-free grammar [29], unlabelled trees [61] and labelled connected graphs [45]. Typically, the counting information is derived from a recurrence relation which also defines a self-reducibility for the structures in question: occasionally, however, a deeper result is involved (e.g., in the case of spanning trees, Kirchoff's matrix tree theorem). We give one more example below, and thereafter in this thesis concentrate on cases where exact counting is apparently not possible.

Generation algorithms which make use of approximate counting information and the correction technique are much less common: a notable exception is the elegant method of Bach [7] for uniformly generating factored integers in a given interval. Another application is a refinement of the scheme proposed by Dixon and Wilf [18] for generating unlabelled graphs. (Their algorithm has polynomially bounded *expected* runtime, but the distribution induced by any polynomial time truncation has very large bias. The refinement guarantees a uniform distribution in polynomial time.) We will not expand on this example here but instead refer the reader to a more elegant attack on the same problem by Wormald [63].

The following little example is chosen partly because it illustrates the reduction to exact counting and partly because of its superficial resemblance to the problem of generating labelled graphs with given vertex degrees, which we shall discuss at length in Chapter 4.

Example 1.11 Consider the relation TREES which associates with each sequence $\mathbf{g} = (g_i)_{i\in I}$ of non-negative integers the set of labelled trees with vertex set I in which vertex i has degree g_i. Clearly, if TREES(g) is non-empty then $g_i > 0$ for all $i \in I$ and $\sum_{i\in I} g_i = 2|I|-2$: we call g *valid* in this case. If g is valid then $g_{i_0} = 1$ for at least one $i_0 \in I$, which suggests a natural self-reducibility for the relation TREES. By considering all possible neighbours of i_0, we may write

$$\text{TREES}(\mathbf{g}) = \bigcup_{j\in I\setminus\{i_0\}} \left\{T \cup \{(i_0,j)\} : T \in \text{TREES}(\mathbf{g}^{(j)})\right\}, \qquad (1.9)$$

where $\mathbf{g}^{(j)}$ is the degree sequence with vertex set $I \setminus \{i_0\}$ satisfying

$$g_i^{(j)} = \begin{cases} g_i - 1, & \text{if } i = j; \\ g_i, & \text{otherwise,} \end{cases}$$

and we have identified trees with their edge sets.

To generate trees on $\mathbf{g}$ uniformly, it is therefore sufficient by Theorem 1.10 to find some efficient means of counting them. This is provided by the following formula, in which $\mathbf{g}$ is assumed valid:

$$\#\text{TREES}(\mathbf{g}) = \frac{(|I|-2)!}{\prod_{i \in I}(g_i - 1)!}.$$

The formula is easily verified by induction on $|I|$ using the recurrence implicit in (1.9) (see, e.g., [39, Problem 4.1]). □

The converse reduction from counting to generation, while perhaps less familiar, is equally intuitive. Clearly, a counting procedure based on random generators cannot be deterministic, so we aim for an efficient randomised approximate counter. Again we make the assumption that R is self-reducible and work with the tree of derivations for a problem instance x with $R(x) \neq \emptyset$. The task at hand is therefore to estimate the number of leaves in a rooted tree T given an almost uniform generator for the leaves in any maximal subtree. For such a subtree S, let $L(S)$ denote the number of leaves in S. The idea is to generate leaves of T and compute the proportion s of this sample which belongs to some subtree S rooted at a child of the root of T. Assuming that s is a reasonable estimate of the ratio $L(S)/L(T)$, we get an approximation for $L(T)$ by recursively estimating $L(S)$ and multiplying the result by s^{-1}.

The quality of the estimate s at each stage depends on the size of the sample and the way in which the subtree S is selected. Clearly, higher accuracy will be achieved if the ratio $L(S)/L(T)$ is large, so we adopt the policy of choosing S so as to maximise the estimate s for the given sample. The following straightforward piece of statistics tells us how large a sample is required in order to achieve a specified accuracy.

Proposition 1.12 *With the above notation, suppose that T has maximum degree d and elements of the sample are generated almost uniformly within tolerance $\xi \in (0,1]$. Then for any $\delta \in (0,1]$, the sample size t required to ensure*

that

$$\Pr\big(s \text{ approximates } L(S)/L(T) \text{ within ratio } 1+2\xi\big) \geq 1-\delta$$

is at most $9d^3(\xi^2\delta)^{-1}$. □

A minor problem arises from the fact that the generator may fail to supply sufficiently many meaningful outputs within a reasonable number of trials. However, if failure occurs in any one trial with probability no more than $1/2$, it is easy to see that

Proposition 1.13 *For any t, the probability that $3t$ trials fail to yield at least t outputs distinct from ? is at most $3/t$.* □

We therefore perform $3t$ trials at each level of recursion and examine the outputs of the first t successful ones: in the unlikely event that insufficient successful outputs are generated, we abort the counting procedure with default value 0.

Suppose now that we wish to approximate $L(T)$ within ratio $1+\epsilon$ using the above method. Let the depth of T be m and assume for the moment that the procedure is not aborted because of generator failure. Setting $\xi = \epsilon/4m$ and $\delta = 1/8m$ in Proposition 1.12, we can fix t so that at each level of the tree the estimate s is accurate within ratio $1+\epsilon/2m$ with probability at least $1-1/8m$. Since there are at most m levels, with probability $(1-1/8m)^m \geq 7/8$ the product of the factors s^{-1} therefore approximates $L(T)$ within ratio $(1+\epsilon/2m)^m \leq 1+\epsilon$ (assuming $\epsilon \leq 1$). Finally, to dispense with generator failure, we assume further that $t \geq 24m$ and deduce from Proposition 1.13 that the procedure runs to completion with probability at least $(1-1/8m)^m \geq 7/8$. The procedure therefore approximates $L(T)$ within ratio $1+\epsilon$ with probability at least $(7/8)^2 \geq 3/4$, as required. Given an oracle which generates leaves almost uniformly within tolerance $\epsilon/4m$, the runtime of the procedure is bounded by $3mt$, which in turn by Proposition 1.12 is bounded by a polynomial in d, m and ϵ^{-1}.

Returning to the case where $T = T_R(x)$ as above, note first that the depth m and degree d of the tree are both polynomially bounded in $|x|$. The generation of leaves in any maximal subtree may be accomplished by an almost uniform generator for R within tolerance $\epsilon/4m$. If the generator is f.p., its runtime on each trial will then be bounded by a polynomial in $|x|$ and ϵ^{-1}, and thus the runtime of the entire procedure will be similarly bounded. The algorithm for

$\Pi := 1$;
while $l_R(x) > 0$ **do begin**
  make $3t$ calls of the form $\mathcal{G}(x, \epsilon/4m)$;
  if at least t of these yield a solution
    then let $Y = \{y_1, \ldots, y_t\}$ be the first t solutions
    else halt with output 0;
  for $w \in \Sigma^{\sigma(z)}$ **do** $s(w) := |\{y \in Y : w \text{ is a prefix of } y\}|/|Y|$;
  let w be such that $s(w) = \max_{w'} s(w')$;
  $\Pi := \Pi/s(w)$; $x := \psi(x, w)$
end;
halt with output Π

Figure 1.3: Procedure for estimating $\#R(x)$

estimating $\#R(x)$ is given in detail in Figure 1.3, in which it is assumed that $\mathcal{G}$ is an almost uniform generator for R.

A partial converse to Theorem 1.10 may now be stated as follows:

Theorem 1.14 (Jerrum, Valiant and Vazirani) *Let R be a self-reducible relation over Σ. If there exists a f.p. almost uniform generator for R then there exists a f.p. randomised approximate counter for R.* □

Combining Theorems 1.10 and 1.14, we arrive at

Corollary 1.15 *For a self-reducible relation R over Σ, the following are equivalent:*

(i) *There exists a f.p. almost uniform generator for R.*

(ii) *There exists a f.p. randomised approximate counter for R.*

(iii) *There exists a polynomially time-bounded randomised approximate counter for R within ratio $1 + O(n^{-k_R})$, where k_R is a constant as above.* □

The implication (iii) $\Rightarrow$ (ii) in Corollary 1.15 indicates that the pair of reductions presented here actually yield a method for bootstrapping a polynomially

time-bounded randomised approximate counter for R within the threshold ratio $1 + O(n^{-k_R})$ to one within ratio $1 + n^{-\beta}$ for *any* desired real β. In view of the hypothesis of Theorem 1.10, however, bootstrapping of less accurate counters is apparently not possible using these techniques.

Unfortunately, the reductions as they stand do not seem to yield a corresponding improvement mechanism for generators. This stems from the fact that the second reduction, unlike the first, demands of its oracle at each level an accuracy which is not simply a function of the local problem instance, but which depends on the *global* quantities ϵ and m: there is no analogue of the correction process employed in generation which would allow larger errors to be handled. Consequently, we have to resort to a generator whose tolerance may be varied.[2] For future reference, however, we state a weaker version of Theorem 1.14 which presupposes only the existence of a generator with fixed tolerance.

Theorem 1.16 *For R and k_R as above, if there exists a polynomially time-bounded almost uniform generator for R within tolerance $O(n^{-k_R})$ then there exists a polynomially time-bounded randomised approximate counter for R within ratio $O(n^{\beta})$ for some constant β.*

Proof Immediate from (1.5) and the proof of Theorem 1.14, with $\xi = c'|x|^{-k_R}$ for some constant c'. □

We shall have a lot more to say about bootstrapping both counters and generators in Chapter 4, where we present a significantly improved version of Theorem 1.10.

1.5 An interesting class of relations

In Section 1.2 it was argued that the counting problem for a p-relation R may be regarded as effectively tractable if there exists a f.p. randomised approximate

---

[2] In [33] it is informally claimed that a generator satisfying the hypothesis of Theorem 1.16 can be bootstrapped to a f.p. almost uniform generator. This turns out to be true (see Theorem 4.8), but the proof relies on the non-trivial machinery developed in Chapter 4.

counter for R. Similarly, we may view the existence of a f.p. almost uniform generator for R as evidence for the tractablility of its uniform generation problem. For the purposes of this discussion, let us call a relation "approximable" if at least one of these problems for it is tractable. In most cases the relation will be self-reducible, so in the light of Section 1.4 if one of the problems is tractable then so is the other.

In the remainder of this thesis, we will make some contribution towards classifying natural relations according to the criterion of approximability, concentrating primarily on techniques for proving positive results. Before embarking on this investigation, however, we should be a little clearer about the kind of relations for which this question is interesting.

Obviously, we are not particularly interested in relations, such as those mentioned in the previous section, whose counting problem is known to be solvable exactly in polynomial time. If the relation is self-reducible, this also implies the existence of a polynomially time-bounded uniform generator for it and the status of both problems is therefore fully resolved. Ideally, we would like some positive evidence of intractablility for the exact counting problem, which usually means #P-completeness. (Note that there are difficult counting problems for which #P-completeness is not an appropriate notion of hardness. This is the case for most of the classical graphical enumeration problems of the form: Given an integer n, compute the number of graphs of size n having a certain property [27]. We refer to these as problems of *unary* type since they have only one input of each size. A more natural concept here is that of $\#\mathrm{P}_1$-completeness [59], which unfortunately has proved rather hard to work with in practice.)

On the other hand, the question of approximablility is also trivial if the existence problem for the relation in question is suspected to be hard, in particular if it is NP-complete. To see this, observe that a polynomially time-bounded almost uniform generator for R within *any* tolerance $\epsilon \geq 0$ immediately yields an efficient solution to the corresponding existence problem in the RP sense defined earlier. If this latter problem were known to be NP-complete, then we could deduce that RP = NP, i.e., that the existence problem for any p-relation is tractable: this is widely held to be almost as unlikely as the assertion that P = NP. The existence of a polynomially time-bounded randomised approximate counter for such a relation within any ratio $\rho \geq 1$ would also imply that RP = NP. By Theorem 1.10, this is immediate if the relation is self-reducible. If

not, we could still use the counter to construct a polynomial time randomised algorithm with bounded *two-sided* error probability for an NP-complete existence problem associated with a self-reducible relation, such as SAT.[3] The error may, however, be made one-sided in polynomial time by exploiting the self-reducibility of SAT: before outputting a "yes" answer, it is first verified by explicit construction of a solution. This would imply that the existence problem for SAT is in RP.

In view of these points, we shall focus on p-relations whose counting problem appears to be hard and whose existence problem is tractable. We have already mentioned that many naturally occurring relations fall into this category: as well as perfect matchings in a graph and satisfying assignments of a DNF formula, examples include matchings (independent sets of edges of any size) in a graph [59], directed trees and s-t-paths in a directed graph [59], and connected spanning subgraphs of a graph [47]. For all these relations, the existence problem lies in P and the counting problem is #P-complete.

Remark We should mention by way of an example a further case in which uniform generation may become trivial. Consider the problem of uniformly generating labelled connected graphs with a given number of vertices. The following naïve procedure provides an efficient solution: simply select a random graph of the required size and output it if it is connected, failing otherwise. The method works since almost every graph of each size is connected (see, e.g., [11]). (A non-trivial "exact" solution to this problem based on the reduction to counting also exists and can be found in [45].) □

Having identified a potentially interesting class of relations, we should check that the question of approximability is a genuine issue for the class: in other words, we should be able to exhibit a natural relation in it which is approximable and another for which even approximate counting and generation are hard. A candidate for the first of these criteria has already been mentioned, namely the DNF satisfiability relation. That this is approximable is shown by Karp and Luby [34], who give a f.p. randomised approximate counter for it. In fact, Jerrum, Valiant and Vazirani [33] prove the slightly stronger result that the relation has a polynomially time-bounded uniform generator. We do not describe

[3] More precisely, we could locate the existence problem for SAT in the class BPP [25].

these algorithms in detail here; the reader may find an alternative approach to this problem in Example 4.4 of Chapter 4.

Conversely, it is possible to find natural relations with easy existence and construction problems which are nevertheless hard to approximate. This complexity gap presumably arises from an implicit requirement in counting and generation that all solutions be "equally accessible". As an example, consider the relation INDSETS which associates with an undirected graph G all independent sets of vertices of G. Note that the construction problem is trivial here since the empty set is always independent. We now proceed to show

Theorem 1.17 *If there exists a polynomially time-bounded uniform generator for* INDSETS *then* RP = NP.

Proof Let IS be the relation which associates with each graph $G = \langle V, E \rangle$ and positive integer k all independent sets of G of size at least k. The existence problem for IS is a standard NP-complete problem [24]: we present a polynomial time reduction from it to the uniform generation problem for INDSETS.

The reduction proceeds by replacing each vertex v of G by a cluster C_v of r independent vertices, where r will be specified below. For each edge (u, v) of G, we add the set of edges $\{(u', v') : u' \in C_u, v' \in C_v\}$, and denote the resulting graph G'.

Now let S be an independent set in G. An independent set S' in G' is a *witness* for S iff it satisfies

$$\{v \in V : C_v \cap S' \neq \emptyset\} = S.$$

Note that every independent set in G' is a witness for a unique set in G. The proof hinges on the fact that, if r is chosen suitably, large independent sets have many more witnesses that small ones. To see this, partition the independent sets of G' into two classes, Large(G, k) and Small(G, k), according to whether they are witnesses for sets in G of size $\geq k$ or $< k$ respectively. Clearly, $|\text{Large}(G, k)| > 0$ iff G contains an independent set of size k. So assume now that this holds, and for $0 \leq l \leq k$ let N_l denote the number of witnesses for each independent set of size l in G. Then we have, from the construction of G',

$$N_{l+1} \geq 2^r N_l \qquad \text{for } 0 \leq l \leq k.$$

Let n be the number of vertices in G. Then G can contain at most $\binom{n}{l}$ independent sets of size l, so

$$|\text{Small}(G,k)| \leq \sum_{l=0}^{k-1} \binom{n}{l} N_l \leq 2^n N_{k-1}\,,$$

since N_l increases with l. Combining the above two inequalities gives

$$|\text{Large}(G,k)| \geq N_k \geq 2^r N_{k-1} \geq 2^{r-n}|\text{Small}(G,k)|\,.$$

Hence by choosing $r = n$ we may ensure that at least half of the independent sets in G' are witnesses for independent sets in G of size k or more. It follows that, if independent sets in G' can be generated uniformly in polynomial time, the existence problem for IS lies in RP, as required. □

Closer examination of the above proof reveals that a polynomially time-bounded almost uniform generator for INDSETS with very large tolerance (any polynomial function of the input size) would also be sufficient. Furthermore, approximate counting even within a large ratio is easily seen to be hard for the same reason.

The above technique of boosting the proportion of solutions which are hard to detect was initially used by Jerrum, Valiant and Vazirani [33] to prove analogous hardness results for the relation of cycles in a directed graph: these transfer directly to undirected graphs and s-t paths (i.e., simple paths between a specified pair of vertices). By parsimonious (or almost parsimonious) reductions from INDSETS, the following relations are also seen to be hard to approximate: all vertex covers and all cliques in a graph , satisfying assignments of a monotone Boolean formula in 2-CNF and 3-colourings of dense[4] graphs. The proof of Theorem 1.17, with minor modifications, shows that the same holds for maximal (i.e., non-extendable) independent sets, and thus also for minimal vertex covers and maximal cliques.

Remark The proof of Theorem 1.17 illustrates a feature of approximate counting of which the reader should be aware. We may view the above reduction

[4]The minimum vertex degree must be at least cn, where n is the number of vertices and $c < 1/2$ is fixed. Interestingly, the counting problem is solvable exactly in polynomial time if $c > 1/2$ [21].

as a proof that the exact counting problem for INDSETS is #P-hard: for with suitable choice of r, by evaluating #INDSETS for the transformed graph G' and looking only at the most significant figures in the output, $\#\text{IS}(G, k)$ can be computed exactly. The latter counting problem is itself trivially #P-complete by parsimonious reduction from SAT. Because only the most significant figures are relevant, the reduction also holds for *approximate* counting. Similarly robust reductions exist for the other relations listed above. The reductions employed in many proofs of #P-hardness, however, appeal to less trivial arithmetic such as polynomial interpolation (see, e.g., [59]) and consequently say nothing about the complexity of approximate counting. □

We conclude our introductory material by quoting some results on the complexity of approximate counting and generation problems for p-relations as a class. From our earlier discussion, it might be expected that these problems are in general intermediate in difficulty between exact counting and existence; it turns out that this can be formalised as follows. Recall that NP is the class of existence problems for p-relations and lies within the first level of the polynomial time hierarchy [53]. Furthermore, the corresponding class #P of exact counting problems is not known to lie within *any* fixed level of the hierarchy. The following result, which is an application of work by Sipser [52] on universal hash functions, indicates that the counting problem for any p-relation may be solved approximately by a machine in the *second* level of the hierarchy, generalised to include randomisation.

Theorem 1.18 (Stockmeyer [54]) *Let $R \subseteq \Sigma^* \times \Sigma^*$ be a p-relation. Then there exists a PTM equipped with an* NP *oracle which is a f.p. randomised approximate counter for R. Furthermore, there exists a deterministic Turing machine equipped with a Σ_2^p oracle which is a f.p. approximate counter for R.* □

As observed in [33], this immediately yields a similar characterisation for generation problems:

Corollary 1.19 *For any p-relation $R \subseteq \Sigma^* \times \Sigma^*$, there exists a PTM with an* NP *oracle which is a f.p. almost uniform generator for R, and a PTM with a Σ_2^p oracle which is a polynomially time-bounded uniform generator for R.*

Proof Observe that the generic p-relation NCOMP defined by

$$\text{NCOMP}(M, x, w) = \{y : M \text{ encodes an NP-machine and } wy \text{ an accepting computation of } M \text{ on input } x\}$$

is self-reducible, and apply Theorem 1.18 and the reduction of Theorem 1.10. □

Note that the class of exactly uniform generation problems for p-relations lies within the *third* level of the hierarchy. By contrast, recall that the corresponding class #P of exact counting problems is not known to lie within *any* fixed level of the hierarchy.

Chapter 2

Markov chains and rapid mixing

This thesis is primarily concerned with positive results on the approximability of natural p-relations, as discussed in Section 1.5. With the exception of certain relations of unary type, where there is a considerable body of work on *analytic* (i.e., closed form) counting estimates, such results are rare in interesting cases. Ideally, we would like to have available some algorithmic paradigms with reasonably wide applicability. To this end, we investigate here a general approach to generation problems based on a simple dynamic stochastic process, namely a finite Markov chain, which moves around a space containing the structures of interest and converges to some desired distribution on them. Since the relations we consider will almost always be self-reducible, the results we obtain carry over directly to the corresponding counting problems by virtue of the observations of Section 1.4.

Techniques of this kind have been in use for some time, particularly in the physical sciences; however, until recently little was known about the non-asymptotic behaviour of Markov chains, so that no rigorous performance guarantees could be given for the resulting algorithms. In this chapter, a new method of analysis for Markov chains is developed which is related to recent research in graph theory on the connection between the subdominant eigenvalues of a graph and its expansion properties. Apart from its inherent interest, this will enable us in later chapters to demonstrate for the first time the existence of efficient approximation algorithms for a number of important counting and generation problems. It will also afford a deeper insight into the nature of the approximations themselves.

2.1 The Markov chain approach to generation problems

Consider the following dynamic stochastic technique for generating objects uniformly at random from a finite set S, assumed very large. Suppose it is possible to construct a Markov chain whose states may be identified with the elements of S. Suppose also that the chain is ergodic with uniform stationary distribution; in other words, if the chain is allowed to evolve for t steps the distribution of the final state approaches the uniform distribution as $t \to \infty$, irrespective of initial state. Then an almost uniform generation procedure for S is obtained by simulating the Markov chain for sufficiently many steps, from some arbitrary initial state, and outputting the element of S corresponding to the final state.

In the following chapters we shall exploit this approach to obtain efficient almost uniform generators for several natural relations as discussed in Chapter 1. More specifically, if R is a relation and x a problem instance with non-empty solution set $R(x)$, the idea is to construct a Markov chain $MC(x)$ some or all of whose states correspond to the structures in $R(x)$. Transitions in the chain will correspond to simple local perturbations of the structures themselves. This technique was recently suggested by Andrei Broder [12] as a means of generating perfect matchings in dense bipartite graphs: in Chapter 3 we will look at this problem in detail and show for the first time that the technique works. The same approach can be used to sample the structures from more general probability distributions by adjusting the stationary distribution of the chain accordingly. Problems of this nature arise frequently in Monte Carlo investigations of physical systems [10], where the states correspond to configurations of the system and appropriate functions of the stationary process to physical constants or parameters. They also lie at the heart of stochastic optimisation methods, such as simulated annealing [37], in which low cost configurations are associated with large weights in the stationary distribution. We shall discuss such applications in more detail in Chapter 3.

Assuming that the local structure of the Markov chain is simple, so that individual transitions can be simulated at small cost, the efficiency of the above procedure depends crucially on the *rate of convergence* of the chain. Specifically, the chain $MC(x)$ should be *rapidly mixing* in the sense that it is close to stationarity after only $p(|x|)$ simulation steps, for some polynomial p. Since the number

of states is in general exponentially large, rapid mixing essentially demands that the chain should lose its memory after visiting only a small fraction of its state space. In applying this technique, we are therefore faced with the problem of deriving *a priori* bounds on the number of simulation steps required to achieve a distribution which is sufficiently close to the limit for the purpose at hand.

In the traditional theory of Markov chains, the question of rate of convergence has received little attention. In recent years, however, it has emerged as an important research area, and a number of analytic techniques have been explored by various authors. Stochastic methods, such as coupling [1], [12] and stopping times [3], are attractive and yield tight bounds for simple chains which possess a highly symmetrical structure. Unfortunately, however, the analysis involved appears to become extremely complex in more interesting cases. In this thesis, we focus on the classical theory based on the eigenvalues of the transition matrix, which seems hitherto to have been of little practical value in non-numerical contexts. Our contribution is to develop from it a simple characterisation of rapid mixing in terms of a structural property of the underlying graph of the chain. This will turn out to be a powerful tool for obtaining useful analytic bounds for a number of interesting chains.

In our development, we will assume that the reader is familiar with the elementary theory of finite Markov chains in discrete time: an introduction can be found in, for example, [23, Chapter XV]. In the remainder of this section, we establish some terminology and notation and quote some basic facts.

Let the sequence of random variables $\left(X_t\right)_{t=0}^{\infty}$ be a time-homogeneous Markov chain on a finite *state space* $[N] = \{0, 1, \ldots, N-1\}$, $N \geq 1$, with *transition matrix* $P = \left(p_{ij}\right)_{i,j=0}^{N-1}$. (Unless otherwise stated, all Markov chains in this thesis will be assumed to be of this form.) Thus for any ordered pair i, j of states the quantity $p_{ij} = \Pr\left(X_{t+1} = j \mid X_t = i\right)$ is the *transition probability* from state i to state j and is independent of t. The matrix P is non-negative and *stochastic*, i.e., its row sums are all unity. For $s \in \mathbb{N}$, the *s-step transition matrix* is simply the power $P^s = \left(p_{ij}^{(s)}\right)$; thus $p_{ij}^{(s)} = \Pr(X_{t+s} = j \mid X_t = i)$, independent of t. We denote the distribution of X_t by the row vector $\boldsymbol{\pi}^{(t)\prime} = \left(\pi_i^{(t)}\right)_{i=0}^{N-1}$, so that $\pi_i^{(t)} = \Pr(X_t = i)$. Here $\boldsymbol{\pi}^{(0)\prime}$ denotes the *initial distribution*, and $\boldsymbol{\pi}^{(t)\prime} = \boldsymbol{\pi}^{(0)\prime} P^t$ for all $t \in \mathbb{N}$. Usually we will have $\pi_i^{(0)} = 1$ for some $i \in [N]$ (and 0 elsewhere); i is then called the *initial state*.

The chain is *ergodic* if there exists a distribution $\pi' = (\pi_i) > 0$ over $[N]$ such that

$$\lim_{s\to\infty} p_{ij}^{(s)} = \pi_j \qquad \forall\, i,j \in [N]\,.$$

In this case, we have that $\pi^{(t)\prime} = \pi^{(0)\prime}P^t \to \pi'$ pointwise as $t \to \infty$, and the limit is independent of $\pi^{(0)\prime}$. The *stationary distribution* π' is the unique vector satisfying $\pi'P = \pi'$, $\sum_i \pi_i = 1$, i.e., the unique normalised left eigenvector of P with eigenvalue 1. Necessary and sufficient conditions for ergodicity are that the chain should be (a) *irreducible*, i.e., for each pair of states $i,j \in [N]$, there is an $s \in \mathbb{N}$ such that $p_{ij}^{(s)} > 0$ (j can be reached from i in a finite number of steps); and (b) *aperiodic*, i.e., $\gcd\{s : p_{ij}^{(s)} > 0\} = 1$ for all $i,j \in [N]$.

Consider now the problem discussed earlier of sampling elements of the state space, assumed very large, according to the stationary distribution π'. The desired distribution can be realised by picking an arbitrary initial state and simulating the transitions of the Markov chain according to the probabilities p_{ij}, which we assume can be computed *locally* as required. As the number t of simulation steps increases, the distribution of the random variable X_t will approach π'. In order to investigate the rate of approach to stationarity, we define the following time-dependent measure of deviation from the limit: for any non-empty subset $U \subseteq [N]$, the *relative pointwise distance* (r.p.d.) over U after t steps is given by

$$\Delta_U(t) = \max_{i,j\in U}\left\{\frac{|p_{ij}^{(t)} - \pi_j|}{\pi_j}\right\}.$$

Thus $\Delta_U(t)$ is just the largest relative difference between $\pi^{(t)\prime}$ and π' at any state $j \in U$, maximised over all possible initial states $i \in U$.[1] The inclusion of the parameter U merely allows us to specify that certain portions of the state space are not relevant in the sampling process, as will prove helpful later. In the case that $U = [N]$, we shall omit the subscript and write simply Δ in place of $\Delta_{[N]}$. The aim of this chapter is to obtain useful bounds on Δ_U as a function of t.

---

[1] We have chosen this measure by analogy with our definition of almost uniform generation in Chapter 1. We could alternatively have used a measure based on the *variation distance*, namely $\Delta'_U(t) = \max_{i\in U}\left\{\sum_j |p_{ij}^{(t)} - \pi_j|\right\}$. For most interesting chains, this choice makes no essential difference to the rapid mixing criterion.

In particular, we want to investigate conditions under which the chains $\mathcal{MC}(x)$ are rapidly mixing in the sense that $\Delta(t)$ becomes very small in polynomial time.

An ergodic Markov chain is said to be *time-reversible* iff either (and hence both) of the following equivalent conditions holds:

(tr1) For all $i, j \in [N]$, $p_{ij}\,\pi_i = p_{ji}\,\pi_j$.

(tr2) The matrix $D^{1/2}PD^{-1/2}$ is symmetric, where $D^{1/2}$ is the diagonal matrix $\mathrm{diag}(\pi_0^{1/2}, \ldots, \pi_{N-1}^{1/2})$ and $D^{-1/2}$ is its inverse.

Condition (tr1) says that in the stationary distribution the expected numbers of transitions per unit time from state i to state j and from state j to state i are equal, and is often referred to as the "detailed balance" property. As we shall see, time-reversible chains are amenable to detailed analysis which is apparently not possible in the general case. For this reason, they play a major rôle in applications where a rigorous quantitative treatment is necessary (see, e.g., [36]).

It is illuminating to identify an ergodic time-reversible chain with a weighted undirected graph (containing self-loops) as follows. The vertex set of the graph is the state space $[N]$ of the chain, and for each pair of states i, j (which need not be distinct) the edge (i, j) has weight $w_{ij} = \pi_i p_{ij} = \pi_j p_{ji}$. By detailed balance, this definition is consistent. Thus there is an edge of non-zero weight between i and j iff $p_{ij} > 0$. We call this graph the *underlying graph* of the chain. It should be clear that such a chain is uniquely specified by its underlying graph.

2.2 Conductance and the rate of convergence

In this section, we establish an intimate relationship between the rate of convergence of an ergodic time-reversible chain and a certain structural property, called the *conductance*, of its underlying graph. Essentially, such a chain will turn out to converge fast iff the conductance is not too small. The crucial step in the proof is a connection between the conductance and the second eigenvalue of the transition matrix of the chain. Similar relationships between subdominant eigenvalues of a graph and a more familiar structural property, known as the *expansion* or *magnification*, have recently been established in a different context by Alon [4] and Alon and Milman [5], and their relevance for the rate of convergence of certain Markov chains noted by Aldous [2].

As already observed, the stationary distribution π' of an ergodic chain is a left eigenvector of P with eigenvalue $\lambda_0 = 1$. Let $\{\lambda_i : 1 \leq i \leq N-1\}$, with $\lambda_i \in \mathbb{C}$, be the remaining eigenvalues (not necessarily distinct) of P. By standard Perron-Frobenius theory for non-negative matrices [50], these satisfy $|\lambda_i| < 1$ for $1 \leq i \leq N-1$. Furthermore, the transient behaviour of the chain, and hence its rate of convergence, is governed by the magnitude of the eigenvalues λ_i. In the time-reversible case, condition (tr2) of the definition implies that the eigenvalues of P are just those of the similar *symmetric* matrix $D^{1/2}PD^{-1/2}$, and so are all real. This fact leads to a clean formulation of the above dependence, expressed in the following pair of propositions.

Proposition 2.1 *Let P be the transition matrix of an ergodic time-reversible Markov chain, π' its stationary distribution and $\{\lambda_i : 0 \leq i \leq N-1\}$ its (necessarily real) eigenvalues, with $\lambda_0 = 1$. Then for any non-empty subset $U \subseteq [N]$ and all $t \in \mathbb{N}$, the relative pointwise distance $\Delta_U(t)$ satisfies*

$$\Delta_U(t) \leq \frac{\lambda_{\max}^t}{\min_{i \in U} \pi_i},$$

where $\lambda_{\max} = \max\{|\lambda_i| : 1 \leq i \leq N-1\}$.

Proof Let $D^{1/2}$ and $D^{-1/2}$ be as in the definition of time-reversibility, so that the matrix $A = D^{1/2}PD^{-1/2}$ is symmetric with the same eigenvalues as P, and these are real. Hence we can select an orthonormal basis $\{\mathbf{e}^{(i)\prime} : 0 \leq i \leq N-1\}$ for $\mathbb{R}^N$ consisting of left eigenvectors of A, where $\mathbf{e}^{(i)\prime} = \left(e_j^{(i)}\right)$ has associated eigenvalue λ_i and $e_j^{(0)} = \pi_j^{1/2}$ for $j \in [N]$.

Following [36], A has the spectral representation

$$A = \sum_{i=0}^{N-1} \lambda_i \mathbf{e}^{(i)} \mathbf{e}^{(i)\prime} = \sum_{i=0}^{N-1} \lambda_i E^{(i)},$$

where $E^{(i)} = \mathbf{e}^{(i)}\mathbf{e}^{(i)\prime}$ is a *dyad* (i.e., has rank 1) with $E^{(i)}E^{(j)} = 0$ for $i \neq j$, and $E^{(i)2} = E^{(i)}$. It follows that, for any $t \in \mathbb{N}$, $A^t = \sum_i \lambda_i^t E^{(i)}$, and hence

$$\begin{aligned} P^t = D^{-1/2}A^tD^{1/2} &= \sum_{i=0}^{N-1} \lambda_i^t (D^{-1/2}\mathbf{e}^{(i)})(\mathbf{e}^{(i)\prime}D^{1/2}) \\ &= \mathbf{1}_N \pi' + \sum_{i=1}^{N-1} \lambda_i^t (D^{-1/2}\mathbf{e}^{(i)})(\mathbf{e}^{(i)\prime}D^{1/2}), \end{aligned}$$

where $\mathbf{1}_N$ is the N-vector all of whose entries are 1; in component form,

$$p_{jk}^{(t)} = \pi_k + \sqrt{\frac{\pi_k}{\pi_j}} \sum_{i=1}^{N-1} \lambda_i^t e_j^{(i)} e_k^{(i)} .$$

By definition, the r.p.d. Δ_U is therefore given by

$$\begin{aligned} \Delta_U(t) &= \max_{j,k \in U} \left\{ \frac{\left| \sum_{i=1}^{N-1} \lambda_i^t e_j^{(i)} e_k^{(i)} \right|}{\sqrt{\pi_j \pi_k}} \right\} \qquad (2.1) \\ &\leq \lambda_{\max}^t \frac{\max_{j,k \in U} \sum_{i=1}^{N-1} |e_j^{(i)}||e_k^{(i)}|}{\min_{j \in U} \pi_j} \\ &\leq \frac{\lambda_{\max}^t}{\min_{j \in U} \pi_j}, \end{aligned}$$

where the second inequality follows from the Cauchy-Schwarz inequality and the orthonormality of the $\mathbf{e}^{(i)}$. □

Proposition 2.2 *With the notation and assumptions of Proposition* 2.1, *the relative pointwise distance* $\Delta(t)$ *over* $[N]$ *satisfies*

$$\Delta(t) \geq \lambda_{\max}^t$$

for all even $t \in \mathbb{N}$. *Moreover, if all eigenvalues of* P *are non-negative, the bound holds for* all $t \in \mathbb{N}$.

Proof The equality (2.1) in the proof of Proposition 2.1 still holds. Setting $U = [N]$ and $k = j$, we see that

$$\Delta(t) \equiv \Delta_{[N]}(t) \geq \max_{j \in [N]} \left\{ \frac{\left| \sum_{i=1}^{N-1} \lambda_i^t e_j^{(i)^2} \right|}{\pi_j} \right\} \geq \lambda_{\max}^t \max_{j \in [N]} \left\{ \frac{e_j^{(i_0)^2}}{\pi_j} \right\}$$

for all (even) natural numbers t, where $\mathbf{e}^{(i_0)}$ is an eigenvector corresponding to an eigenvalue of modulus $\lambda_{\max}$. By orthonormality of $\mathbf{e}^{(0)}$ and $\mathbf{e}^{(i_0)}$ and the form of $\mathbf{e}^{(0)}$, this latter quantity is bounded below by $\lambda_{\max}^t$ as required. □

Propositions 2.1 and 2.2 say that, provided π' is not extremely small in any state of interest, the convergence of a time-reversible chain will be rapid in the sense indicated earlier iff $\lambda_{\max}$ is suitably bounded away from 1. The first of these conditions can be checked immediately from our knowledge of π', and is rarely violated in practice. We therefore focus our attention on the second condition, which is not so easily handled. (Recall that P is assumed to be a large matrix, so that direct numerical evaluation of the eigenvalues is not feasible.)

Suppose that the eigenvalues of P are ordered so that $1 = \lambda_0 > \lambda_1 \geq \cdots \geq \lambda_{N-1} > -1$. Then the value of $\lambda_{\max}$ is governed by λ_1 and λ_{N-1}, the latter being significant only if some of the eigenvalues are negative. Negative eigenvalues in fact present no essential obstacle to rapid convergence because the chain can always be modified in such a way that the stationary distribution is preserved and all eigenvalues become non-negative without risk of slowing down the convergence too much. Note that this simply corresponds to eliminating oscillatory behaviour and is achieved by uniformly boosting the self-loop probabilities, as described next.

Proposition 2.3 *With the notation of Proposition 2.1, suppose also that the eigenvalues of P are ordered so that $1 = \lambda_0 > \lambda_1 \geq \cdots \geq \lambda_{N-1} > -1$, let $p_{\min} = \min_{i \in [N]} p_{ii}$, and define the stochastic matrix P' by*

$$P' = \begin{cases} P, & \text{if } p_{\min} \geq 1/2; \\ \alpha P + (1-\alpha) I_N, & \text{otherwise, where } \alpha = \big(2(1-p_{\min})\big)^{-1}. \end{cases}$$

(I_N is the $N \times N$ identity matrix.) Then the modified chain with transition matrix P' is also ergodic and time-reversible with stationary distribution π', and its eigenvalues $\{\lambda'_i\}$, similarly ordered, satisfy $\lambda'_{N-1} \geq 0$, $\lambda'_{\max} = \lambda'_1 \leq \frac{1}{2}(1 + \lambda_1)$.

Proof Suppose first that $p_{\min} \geq 1/2$, and consider the non-negative matrix $2P - I_N$. This is clearly stochastic and irreducible (though not necessarily aperiodic), and has eigenvalues $\mu_i = 2\lambda_i - 1$. By Perron-Frobenius, $\mu_i \geq -1$ for all $i \in [N]$, which implies that $\lambda'_{N-1} = \lambda_{N-1} \geq 0$.

If on the other hand $p_{\min} < 1/2$, the matrix $P' = \alpha P + (1-\alpha) I_N$ is non-negative and has $\min_i p'_{ii} = 1/2$, so as above all its eigenvalues λ'_i are non-negative. But clearly $\lambda'_i = \alpha\lambda_i + (1-\alpha)$ for all $i \in [N]$, so we must have $\lambda'_{\max} = \lambda'_1 \leq \frac{1}{2}(1 + \lambda_1)$.

□

We turn now to the more substantial problem of bounding the *second eigenvalue* λ_1 away from 1. We shall do this by relating λ_1 to a more accessible structural property of the underlying graph.

Intuitively, we would expect an ergodic chain to converge rapidly if it is unlikely to "get stuck" in any subset S of the state space whose total stationary probability is fairly small. We can formalise this idea by considering the cut edges which separate S from the rest of the space in the underlying graph, and stipulating that these must be capable of supporting a sufficiently large "flow" in the graph, viewed as a network. With this in mind, for any non-empty subset S of states with non-empty complement $\overline{S}$ in $[N]$ we define the quantity $\Phi_S = F_S/C_S$, where

$$C_S = \sum_{i \in S} \pi_i \qquad \text{the } \textit{capacity} \text{ of } S;$$

$$F_S = \sum_{\substack{i \in S \\ j \in \overline{S}}} p_{ij}\,\pi_i \qquad \text{the } \textit{ergodic flow} \text{ out of } S.$$

Note that $0 \leq F_S \leq C_S < 1$. Φ_S may be visualised as the conditional probability that the stationary process crosses the cut from S to $\overline{S}$ in a single step, given that it starts in S. Finally, we define the global *conductance* of the chain by

$$\Phi = \min_{\substack{0<|S|<N \\ C_S \leq 1/2}} \Phi_S\,.$$

It is easy to see that $F_S = F_{\overline{S}}$ for all such sets S. This implies that $\Phi_{\overline{S}} = \Phi_S\big(C_S/(1-C_S)\big)$, so we may equivalently write

$$\Phi = \min_{0<|S|<N} \max\{\Phi_S, \Phi_{\overline{S}}\}\,.$$

Now suppose that the chain is time-reversible, and let G be its underlying graph. Then for all S as above we have

$$F_S = F_{\overline{S}} = \sum_{\substack{i \in S \\ j \in \overline{S}}} w_{ij}\,,$$

a function of the edge weights of G. The conductance $\Phi \equiv \Phi(G)$ may then be viewed as a structural property of the weighted graph G. In view of the above remarks, we might hope that $\Phi(G)$, which in some sense measures the minimum relative connection strength between "small" subsets S and the rest of the space, is related to the rate of convergence of the chain. This relationship is manifested via two separate bounds on the second eigenvalue λ_1.

Lemma 2.4 *For an ergodic time-reversible Markov chain with underlying graph G, the second eigenvalue λ_1 of the transition matrix satisfies*

$$\lambda_1 \leq 1 - \frac{\Phi(G)^2}{2}.$$

Proof Let $\mathbf{e}' = (e_i)_{i=0}^{N-1}$ be an eigenvector of P with associated eigenvalue $\lambda < 1$, and define the matrix $Q = I_N - P$ (the "Laplace operator" associated with P). Then clearly

$$\mathbf{e}'Q = (1-\lambda)\mathbf{e}'. \tag{2.2}$$

Define the subset of states $S = \{i \in [N] : e_i > 0\}$. Since P is stochastic and $\lambda < 1$, it follows that $\sum_i e_i = 0$. Hence $0 < |S| < N$, and we may assume without loss of generality that $C_S = \sum_{i\in S} \pi_i \leq 1/2$. Now let $\hat{\mathbf{e}}' = (\hat{e}_i)$ be the vector defined by

$$\hat{e}_i = \begin{cases} e_i/\pi_i, & \text{for } i \in S; \\ 0, & \text{otherwise.} \end{cases}$$

Renumbering states as necessary, we shall assume that $\hat{e}_0 \geq \hat{e}_1 \geq \cdots \geq \hat{e}_{N-1}$, which implies also that $S = \{0, 1, \ldots, r\}$ for some r with $0 \leq r < N-1$.

Taking the inner product of (2.2) with $\hat{\mathbf{e}}'$ gives

$$(\mathbf{e}'Q, \hat{\mathbf{e}}') = (1-\lambda)(\mathbf{e}', \hat{\mathbf{e}}'). \tag{2.3}$$

The right-hand side of (2.3) is just

$$(1-\lambda)\sum_{i\in S} \pi_i \hat{e}_i^2. \tag{2.4}$$

Note that if $Q = (q_{ij})$ then $q_{ij} = -p_{ij}$ for $i \neq j$, and $q_{ii} = 1 - p_{ii} = \sum_{j\neq i} p_{ij}$, so we can expand the left-hand side of (2.3) as

$$\begin{aligned} \sum_{i\in S}\sum_{j\in[N]} \hat{e}_i q_{ji} e_j &\geq \sum_{i\in S}\sum_{j\in S} \hat{e}_i q_{ji} e_j \\ &= -\sum_{i\in S}\sum_{\substack{j\in S \\ j\neq i}} w_{ij}\hat{e}_i\hat{e}_j + \sum_{i\in S}\sum_{j\neq i} w_{ij}\hat{e}_i^2 \\ &= -2\sum_{i<j} w_{ij}\hat{e}_i\hat{e}_j + \sum_{i<j} w_{ij}(\hat{e}_i^2 + \hat{e}_j^2) \\ &= \sum_{i<j} w_{ij}(\hat{e}_i - \hat{e}_j)^2, \end{aligned} \tag{2.5}$$

where the inequality follows from the fact that all contributions with $j \notin S$ are positive. Using (2.4) and (2.5), equation (2.3) therefore yields

$$1 - \lambda \geq \frac{\sum_{i<j} w_{ij}(\hat{e}_i - \hat{e}_j)^2}{\sum_{i \in S} \pi_i \hat{e}_i^2}. \tag{2.6}$$

Now consider the sum

$$\sum_{i<j} w_{ij}(\hat{e}_i + \hat{e}_j)^2 \leq 2 \sum_{i<j} w_{ij}(\hat{e}_i^2 + \hat{e}_j^2) \leq 2 \sum_{i \in S} \pi_i \hat{e}_i^2 .$$

Combining this with (2.6) gives

$$\begin{aligned} 1 - \lambda &\geq \frac{\sum_{i<j} w_{ij}(\hat{e}_i - \hat{e}_j)^2}{\sum_{i \in S} \pi_i \hat{e}_i^2} \cdot \frac{\sum_{i<j} w_{ij}(\hat{e}_i + \hat{e}_j)^2}{2 \sum_{i \in S} \pi_i \hat{e}_i^2} \\ &\geq \frac{1}{2} \left(\frac{\sum_{i<j} w_{ij}(\hat{e}_i^2 - \hat{e}_j^2)}{\sum_{i \in S} \pi_i \hat{e}_i^2} \right)^2 , \end{aligned} \tag{2.7}$$

where we have used the Cauchy-Schwarz inequality. To complete the proof, we need to relate the quotient in (2.7) to the quantity $\Phi(G)$.

To do this, consider the increasing sequence $\left(S_k\right)_{k=0}^{r}$ of subsets of S with $S_k = \{0, \ldots, k\}$. The numerator of the quotient in (2.7) may be expressed in terms of ergodic flows across the boundaries between successive sets S_k as follows:

$$\begin{aligned} \sum_{i<j} w_{ij}(\hat{e}_i^2 - \hat{e}_j^2) &= \sum_{i<j} w_{ij} \sum_{i \leq k < j} (\hat{e}_k^2 - \hat{e}_{k+1}^2) \\ &= \sum_{k=0}^{r} (\hat{e}_k^2 - \hat{e}_{k+1}^2) \sum_{\substack{i \in S_k \\ j \notin S_k}} w_{ij} \\ &= \sum_{k=0}^{r} (\hat{e}_k^2 - \hat{e}_{k+1}^2) F_{S_k} . \end{aligned} \tag{2.8}$$

Now the capacities of the S_k satisfy $C_{S_k} \leq C_S \leq 1/2$ for $0 \leq k \leq r$, and hence by definition of Φ, $F_{S_k} \geq \Phi(G)\, C_{S_k}$. We therefore get from (2.8)

$$\begin{aligned} \sum_{i<j} w_{ij}(\hat{e}_i^2 - \hat{e}_j^2) &\geq \Phi(G) \sum_{k=0}^{r} (\hat{e}_k^2 - \hat{e}_{k+1}^2) C_{S_k} \\ &= \Phi(G) \sum_{k=0}^{r} (\hat{e}_k^2 - \hat{e}_{k+1}^2) \sum_{i=0}^{k} \pi_i \\ &= \Phi(G) \sum_{i=0}^{r} \pi_i \sum_{k=i}^{r} (\hat{e}_k^2 - \hat{e}_{k+1}^2) \\ &= \Phi(G) \sum_{i \in S} \pi_i \hat{e}_i^2 . \end{aligned}$$

This inequality ensures that the quotient in (2.7) is bounded below by $\Phi(G)$, so that finally

$$1 - \lambda \geq \frac{\Phi(G)^2}{2},$$

as required. □

Combining Propositions 2.1 and 2.3 and Lemma 2.4, we arrive at the first major result of this section, namely an upper bound on the distance from stationarity of a time-reversible chain in terms of the conductance of its underlying graph.

Theorem 2.5 *Let G be the underlying graph of an ergodic time-reversible Markov chain all of whose eigenvalues are non-negative, and π its stationary distribution. Then for any non-empty subset $U \subseteq [N]$ and all $t \in \mathbb{N}$, the relative pointwise distance $\Delta_U(t)$ satisfies*

$$\Delta_U(t) \leq \frac{\left(1 - \Phi(G)^2/2\right)^t}{\min\limits_{i \in U} \pi_i}. \qquad \square$$

Remarks (a) In the interests of simplicity, Theorem 2.5 is stated only for chains with non-negative eigenvalues. Of course, Proposition 2.3 tells us that any chain can be modified in a crude way so that this condition holds: the effect of this operation on the conductance is to reduce it by at most a factor of $1/2$. In practice it may often be possible to reason about negative eigenvalues on an *ad hoc* basis for the chain at hand. Proposition 2.1 and Lemma 2.4 may then be used directly to get a bound on $\Delta_U(t)$.

(b) Theorem 2.5 says that $\Delta(t)$ is bounded above by a function of the form $2^{-(t-t_0)/\beta}$, where β and t_0 are determined by $\Phi(G)$ and π. Thus $\Delta(t)$ is guaranteed to decrease at an exponential rate after an initial "delay" of $t_0 = c_1 \Phi(G)^{-2} \lg \pi_{\min}^{-1}$ time units, where $\pi_{\min} = \min_i \pi_i$ and c_1 is a constant. Once this regime sets in, the actual exponential rate of convergence is governed by the "time constant" $\beta = c_2 \Phi(G)^{-2}$, for some constant c_2. The number of steps required to ensure a r.p.d. of ϵ is at most $t_0 + \beta \lg \epsilon^{-1}$. We shall distil the essential features of this behaviour for our purposes into Corollary 2.8 of the next section. □

Our next aim is to derive a partial converse of Theorem 2.5. First we require a bound on λ_1 complementary to that of Lemma 2.4.

Lemma 2.6 *For an ergodic time-reversible Markov chain with underlying graph G, the second eigenvalue λ_1 of the transition matrix satisfies*

$$\lambda_1 \geq 1 - 2\Phi(G).$$

Proof As in the proof of Lemma 2.4, we work with the matrix $Q = I_N - P$, whose eigenvalues are $\{1 - \lambda_i\}$ for λ_i an eigenvalue of P. First we get a general bound on λ_1 using a variational principle.

Define $B = D^{1/2}QD^{-1/2}$; B is symmetric by virtue of time-reversibility, and its eigenvalues are those of Q. Furthermore, if $\pi' = (\pi_i)$ is the stationary distribution of P, then the vector $\mathbf{e}' = (e_i)$ with $e_i = \pi_i^{1/2}$ is a left eigenvector of B with eigenvalue 0. Now let $\mathbf{g}' = (g_i)$ be any vector orthogonal to $\mathbf{e}'$. Since B is symmetric with second smallest eigenvalue $1 - \lambda_1$, the classical calculus of variations tells us that

$$(1 - \lambda_1)(\mathbf{g}', \mathbf{g}') \leq (\mathbf{g}'B, \mathbf{g}'). \tag{2.9}$$

To relate this to Q, introduce the vectors $\mathbf{f}' = (f_i) = \mathbf{g}'D^{1/2}$ and $\hat{\mathbf{f}}' = (\hat{f}_i) = \mathbf{f}'D^{-1}$. Then (2.9) becomes

$$(1 - \lambda_1)(\mathbf{f}', \hat{\mathbf{f}}') \leq (\mathbf{f}'Q, \hat{\mathbf{f}}'), \tag{2.10}$$

and this holds for all $\mathbf{f}'$ with $\sum_i f_i = 0$. The right-hand side of (2.10) can be rewritten along similar lines to the proof of Lemma 2.4: it is a simple matter to check that

$$\begin{aligned}(\mathbf{f}'Q, \hat{\mathbf{f}}') &= \sum_{i \in [N]} \sum_{j \in [N]} f_i q_{ij} \hat{f}_j \\ &= -2 \sum_{i<j} \hat{f}_i \hat{f}_j w_{ij} + \sum_i \hat{f}_i^2 \pi_i q_{ii} \\ &= \sum_{i<j} w_{ij} (\hat{f}_i - \hat{f}_j)^2, \end{aligned} \tag{2.11}$$

where as usual $w_{ij} = \pi_i p_{ij} = \pi_j p_{ji}$ denotes the weight of the edge (i, j) in G.

Now let S be any subset of states for which $C_S \leq 1/2$. The idea is to select a particular vector $\mathbf{f}'$ for which (2.10) yields a good bound on Φ_S. With this in mind, set

$$f_i = \begin{cases} \dfrac{\pi_i}{C_S}, & \text{for } i \in S; \\ \dfrac{-\pi_i}{(1 - C_S)}, & \text{for } i \notin S. \end{cases}$$

Then clearly $\sum_i f_i = 0$, so (2.10) holds for $\mathbf{f}$. Furthermore, we have

$$(\mathbf{f}', \hat{\mathbf{f}}') = \sum_{i \in [N]} \frac{f_i^2}{\pi_i} = \sum_{i \in S} \frac{\pi_i}{C_S^2} + \sum_{i \notin S} \frac{\pi_i}{(1 - C_S)^2} = \left(\frac{1}{C_S} + \frac{1}{1 - C_S} \right),$$

and from (2.11), since $\hat{\mathbf{f}}'$ is constant on S and $\overline{S}$,

$$(\mathbf{f}'Q, \mathbf{f}) = \sum_{\substack{i \in S \\ j \notin S}} w_{ij} \left(\frac{1}{C_S} + \frac{1}{1 - C_S} \right)^2 = F_S \left(\frac{1}{C_S} + \frac{1}{1 - C_S} \right)^2 .$$

Inequality (2.10) therefore yields

$$1 - \lambda_1 \leq F_S \left(\frac{1}{C_S} + \frac{1}{1 - C_S} \right) \leq 2 \frac{F_S}{C_S} = 2\Phi_S .$$

Since S was chosen arbitrarily, we have the bound

$$\lambda_1 \geq 1 - 2\Phi(G),$$

which completes the proof. □

Proposition 2.2 and Lemma 2.6 together yield a converse of Theorem 2.5.

Theorem 2.7 *Let G be the underlying graph of an ergodic time-reversible Markov chain all of whose eigenvalues are non-negative, and suppose that $\Phi(G) \leq 1/2$. Then the relative pointwise distance $\Delta(t)$ over $[N]$ satisfies*

$$\Delta(t) \geq \left(1 - 2\Phi(G)\right)^t$$

for all $t \in \mathbb{N}$. □

Remarks (a) For simplicity, and in parallel with Theorem 2.5, we have stated Theorem 2.7 only for chains with non-negative eigenvalues and small conductance. If the modification procedure of Proposition 2.3 is applied to an arbitrary chain then both of these conditions are guaranteed to hold. (Note in particular that if $p_{\min} \geq 1/2$ then $\Phi_S \leq 1/2$ for any non-empty subset S.) For chains with negative eigenvalues, the bound of Theorem 2.7 holds for all *even* $t \in \mathbb{N}$, by Proposition 2.2.

(b) Lemmas 2.4 and 2.6 are closely related to recent work of Alon [4] and Alon and Milman [5], in which a relationship between a similar structural property of

(simple, unweighted) graphs and the second eigenvalue of the adjacency matrix is established; indeed, restricted versions of our bounds follow directly from their results (see (c) below). This property, which they call the *magnification*, measures the minimum number of vertices adjacent to a small subset S as a fraction of $|S|$, and is a generalisation of the widely-studied concept of expansion for bipartite graphs. Our conductance Φ is a weighted edge analogue of magnification, and is a more natural quantity to study in the present application.

There is another important difference in emphasis between our approach and that of [4] and [5]. In the latter, one is interested in certifying the magnification properties of a given graph by computing its eigenvalues numerically. Here we are working with very large graphs arising from time-reversible Markov chains where explicit evaluation of the eigenvalues is not feasible. However, the special structure of the graphs may allow us to derive bounds on Φ *analytically*, which by Lemmas 2.4 and 2.6 translate into bounds on λ_1.

(c) The significance of Alon's result [4] as a sufficient condition for rapid convergence of certain Markov chains has been observed by other authors (see, e.g., [2], [13], [46]). In particular, Aldous [2] states a restricted form of Theorem 2.5 for random walks on regular simple graphs. However, our characterisation based on the conductance provides a considerably cleaner and more general formulation of this connection.

(d) We should also point out that Lemma 2.4 parallels an earlier continuous result of Cheeger [15] for Riemannian manifolds. A similar result for infinite graphs has been obtained by Dodziuk [19]. □

2.3 A characterisation of rapid mixing

We conclude this chapter by giving a precise definition of the rapid mixing property mentioned informally earlier, and using the results of the previous section to characterise it in terms of the conductance. We will also point out certain limitations of this characterisation.

Suppose that we have a *family* of ergodic Markov chains $\mathcal{MC}(x)$ parameterised on strings $x \in \Omega \subseteq \Sigma^*$. For each $x \in \Omega$, let $\Delta^{(x)}(t)$ denote the r.p.d. of $\mathcal{MC}(x)$

```
(1)   if R(x) = ∅ then halt with output ?
      else begin
(2)     state := initial state of MC(x);
(3)     for t := 1 to q(|x|, lg (2ε)^{-1}) do
            simulate one step of MC(x);
(4)     if state ∈ R(x) then halt with output state
(5)                     else halt with output ?
      end
```

Figure 2.1: Markov chain simulation paradigm

over its entire state space after t steps, and define the function $\tau^{(x)} : \mathbb{R}^+ \to \mathbb{N}$ by

$$\tau^{(x)}(\epsilon) = \min\left\{t \in \mathbb{N} : \Delta^{(x)}(t') \leq \epsilon \text{ for all } t' \geq t\right\} .$$

We call such a family *rapidly mixing* iff there exists a polynomially bounded function $q : \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ such that

$$\tau^{(x)}(\epsilon) \leq q(|x|, \lg \epsilon^{-1})$$

for all $x \in \Omega$ and $0 < \epsilon \leq 1$. The idea is that simulation of a rapidly mixing family should provide an efficient sampling scheme for the associated stationary distributions.

Clearly, the above definition is meaningful only when the string x is an implicit description of the corresponding chain, in some suitable sense. In the applications we have in mind here, x will be a problem instance and the state space of $MC(x)$ will include the solution set $R(x)$ of some p-relation R. As we are only interested in non-empty solution sets, we have $\Omega = \{x \in \Sigma^* : R(x) \neq \emptyset\}$.

For future reference, we now spell out the Markov chain simulation paradigm for generating solutions of R (see Figure 2.1). Here $\langle x, \epsilon \rangle \in \Sigma^* \times \mathbb{R}^+$ denotes the input, where we assume without loss of generality that $\epsilon \leq 1$, and q is some function (not necessarily polynomially bounded) such that $q(|x|, \lg \epsilon^{-1}) \geq \tau^{(x)}(\epsilon)$. In line (2) the initial state may be chosen arbitrarily by any convenient means. If the stationary distribution of each chain $MC(x)$ is uniform over $R(x)$, then the algorithm of Figure 2.1 yields a f.p. almost uniform generator for R under the following conditions:

(mc1) The construction problem for R is solvable in polynomial time (so that non-emptiness of $R(x)$ can be tested and an initial state found).

(mc2) Individual steps of $\mathcal{MC}(x)$ can be simulated by an OCM in time polynomial in $|x|$.

(mc3) In the stationary distribution of $\mathcal{MC}(x)$, the probability of being at an element of $R(x)$ is bounded below by $1/p(|x|)$ for some polynomial p (so that the probability of failure in line (5) is not too large, and can be reduced to $1/2$ by polynomially many repeated trials).

(mc4) q is polynomially bounded, i.e., the family is rapidly mixing. (Note that, since $\epsilon \leq 1$, a r.p.d. of $\epsilon/2$ ensures a tolerance of at most ϵ.)

In the examples we consider, conditions (mc1) and (mc2) will always be satisfied: we have already discussed reasons why (mc1) should hold, while (mc2) expresses the idea that the stochastic process has a simple local structure. Condition (mc3) merely says that auxiliary states may be included for convenience provided that the solutions themselves are not relegated to an insignificant portion of the state space: we shall meet examples later where this flexibility is useful. The key criterion is therefore (mc4), that of rapid mixing. As we have already observed, since the state space may be exponentially large as a function of $|x|$, rapid mixing demands in general that the chains are very close to stationarity after visiting only a small proportion of their states. (The definition of rapid mixing is sometimes formulated directly in this way when the state spaces of the chains $\mathcal{MC}(x)$ have a highly symmetrical structure [1].)

Assume now that all chains in some family are time-reversible and are known to have non-negative eigenvalues; the latter can always be arranged via the modification procedure of Proposition 2.3. Furthermore, suppose that the minimum probability $\pi_{\min}^{(x)}$ assigned to any state in the stationary distribution of $\mathcal{MC}(x)$ satisfies

$$\lg \pi_{\min}^{(x)} \geq -q'(|x|) \tag{2.12}$$

for all $x \in \Omega$ and some polynomial q'. Then the results of the previous section imply the following:

Corollary 2.8 *Let $\{\mathcal{MC}(x) : x \in \Omega\}$ be a family of ergodic time-reversible Markov chains, and let $G(x)$ be the underlying graph of $\mathcal{MC}(x)$. Under the above*

assumptions, the family is rapidly mixing iff

$$\Phi(G(x)) \geq 1/p(|x|)$$

for all $x \in \Omega$ and some polynomial p.

Proof If $\Phi(G(x)) \geq 1/p(|x|)$, then it is easy to see from Theorem 2.5 that, for all $x \in \Omega$ and $0 < \epsilon \leq 1$,

$$\tau^{(x)}(\epsilon) \leq 2p(|x|)^2 \ln\left(1/\epsilon\pi^{(x)}_{\min}\right).$$

We may therefore take $q(n,\delta) = 2p(n)^2(q'(n) + \delta)$ in the definition of rapid mixing: this is clearly polynomially bounded as required.

Conversely, suppose the rapid mixing property holds. Then there exists a polynomial $\tilde{q}$ such that $\tau^{(x)}(1/2) \leq \tilde{q}(|x|)$ for all $x \in \Omega$. By Theorem 2.7 we must therefore have $\Phi(G(x)) \geq 1/8\tilde{q}(|x|)$, since otherwise

$$\Delta^{(x)}\left(\tilde{q}(|x|)\right) \geq \left(1 - 2\Phi(G(x))\right)^{\tilde{q}(|x|)} > 1 - 1/4 = 3/4.$$

Hence the conductance must be polynomially bounded below. This completes the proof. □

Corollary 2.8 provides a partial characterisation of the rapid mixing property in terms of the conductance of the underlying graphs of the family. The characterisation is incomplete because of the non-negativity assumption on the eigenvalues and the lower bound (2.12) imposed on $\pi^{(x)}_{\min}$. We can justify the restriction to chains having non-negative eigenvalues on the grounds that our ultimate aim is to construct efficient sampling procedures for a given family of distributions. The modification procedure of Proposition 2.3 is effective and cannot destroy the rapid mixing property of the original family: it is therefore reasonable to incorporate it into the chains we consider. The following example confirms that, for chains with negative eigenvalues, rapid mixing is *not* guaranteed by a polynomial lower bound on the conductance.

Example 2.9 Consider the family of chains $\mathcal{MC}(n)$ parameterised on natural numbers $n \in \mathbb{N}^+$, in which $\mathcal{MC}(n)$ has state space $V_1 \cup V_2$ for disjoint V_1, V_2 with $|V_1| = |V_2| = N/2$ and $N = 2^n$. The transition probabilities are given by

$$p_{ij} = \begin{cases} 2(1-\alpha)/N & \text{if } i \in V_1,\ j \in V_2 \text{ or } i \in V_2,\ j \in V_1; \\ \alpha & \text{if } i = j; \\ 0 & \text{otherwise,} \end{cases}$$

where α will be specified below. The chain $\mathcal{MC}(n)$ is obviously ergodic for $0 < \alpha < 1$; since it is symmetric, it is also time-reversible with uniform stationary distribution.

To calculate $\Phi(G(n))$, consider any non-empty subset $S = A \cup B$ of states, with $A \subseteq V_1$, $B \subseteq V_2$ and $|S| \leq N/2$. Writing $a = |A|/N$ and $b = |B|/N$, we have $C_S = a + b$ and

$$F_S = \frac{2(1-\alpha)}{N}\left(a(N/2 - bN) + b(N/2 - aN)\right).$$

It follows that

$$\Phi_S \equiv \frac{F_S}{C_S} = (1-\alpha)\left(1 - \frac{4ab}{a+b}\right).$$

The latter quantity is minimised under the constraint $a + b \leq 1/2$ at the values $a = b = 1/4$, whence

$$\Phi(G(n)) = \min_S \Phi_S = (1-\alpha)/2. \tag{2.13}$$

Now consider the probability that the chain, when started from some initial state in V_1, will be found in V_1 at time $t = 2m + 1$ for $m \in \mathbb{N}$. This can happen only if at least one self-loop is traversed during this period, so the probability is at most $1 - (1-\alpha)^t$. If we set $\alpha = N^{-1}$ (say), then if t is permitted to grow only polynomially with n this probability tends to 0 as $n \to \infty$. It follows that the family is not rapidly mixing. Equality (2.13), however, ensures that the conductance is large.

Note that the above chains fail to converge fast because they are "almost periodic". Informally, the chain $\mathcal{MC}(n)$ can be viewed as a pair of processes on V_1 and V_2 respectively, each of which converges fast but which hardly communicate owing to the small value of α. The modification procedure of Proposition 2.3 makes the chains rapidly mixing by increasing the value of α to $1/2$. □

Next we consider the effect of the stationary distribution on the characterisation of Corollary 2.8. The following pair of examples demonstrate that the assumption of a lower bound of the form (2.12) on $\pi^{(x)}_{\min}$ is necessary, since families with very small stationary state probabilities may exhibit a range of convergence behaviour when $\Phi(G(x))$ is large.

Example 2.10 For $n \in \mathbb{N}^+$, let $\mathcal{MC}(n)$ be a one-dimensional geometric random walk on state space $[N]$, where $N = 2^n$, with transition probabilities $p_{i(i+1)} =$

$1/3$, $p_{i(i-1)} = 2/3$ for $1 \leq i \leq N-2$, and $p_{00} = p_{(N-1)(N-1)} = 2/3$, $p_{01} = p_{(N-1)(N-2)} = 1/3$. (Thus states 0 and $N-1$ form reflecting barriers.) Then $\mathcal{MC}(n)$ is ergodic and time-reversible with stationary distribution $\pi_i = 2^{-i-1}$ for $0 \leq i \leq N-2$ and $\pi_{(N-1)} = 2^{-N+1}$. Furthermore, the conductance $\Phi(G(n))$ is exactly $1/3$, as may readily be verified. Hence, if the chain is modified as in Proposition 2.3 to eliminate negative eigenvalues, the conductance is still $1/6$. However, this family is not rapidly mixing since the number of states reachable from any initial state in t steps is at most $2t+1$. This does not contradict Corollary 2.8 since $\lg \pi_{\min}^{(n)} = -2^n + 1$. □

Example 2.11 To see that small stationary state probabilities do not in themselves preclude the possibility of rapid mixing, for $n \in \mathbb{N}^+$ let the chain $\mathcal{MC}(n)$ have state space $[N]$ with $N = 2^n$. Define transition probabilities by

$$p_{ij} = \begin{cases} \alpha & \text{if } j = 0; \\ (1-\alpha)/(N-1) & \text{otherwise.} \end{cases}$$

This chain is clearly ergodic for $0 < \alpha < 1$; the stationary distribution is given by $\pi_0 = \alpha$ and $\pi_i = (1-\alpha)/(N-1)$ for $1 \leq i \leq N-1$, and time-reversibility follows. Note that $P = \mathbf{1}_N \pi'$, i.e., P is idempotent. This implies that $\Delta^{(x)}(t) = 0$ for all $t \geq 1$, so the chain attains its limit in a single step from any initial state. Moreover, all eigenvalues of P other than λ_0 are equal to zero. Since the above holds for any value of α, we can arrange for $\lg \pi_{\min}^{(n)} = \lg \alpha = -2^n$ without affecting the rapid mixing property. □

Fortunately, pathological cases such as those described in the examples above rarely occur naturally. Moreover, states with extremely small weight in the stationary distribution are typically not relevant to the sampling process, and their effect can be eliminated by working with $\Delta_U(t)$ for some suitably chosen subset U of states. Hence for most practical purposes the conductance may be taken as a reliable characterisation of rapid mixing for time-reversible families.

In the sequel, extensive use will be made of the positive part of the above characterisation to show that certain natural families are rapidly mixing, thus providing efficient sampling schemes for the associated stationary distributions. We shall see that, for several chains with a rather complex structure, the conductance may be quite accessible while the rate of convergence is apparently not easily investigated by other means. Thus we contend that the characterisation of

rapid mixing presented in this chapter is a potentially powerful tool for analysing the transient behaviour of a much wider class of Markov chains than has hitherto been possible.

We close this chapter with an observation which is frequently useful in applications. Many natural Markov chains can be viewed as a simple random walk on a graph $H = (V, E)$ in which transitions are made from any vertex v to an adjacent vertex with probability β/d, where d is the maximum vertex degree of H and $\beta \leq 1$ is a positive constant. In addition, v has self-loop probability $1 - \beta \deg(v)/d$, where $\deg(v)$ is the degree of v in H. If H is connected and $\beta < 1$ then the chain is certainly ergodic; symmetry ensures that it is time-reversible with uniform stationary distribution. Note that in the underlying graph all edges of H have equal non-zero weight, while all remaining edges other than self-loops have weight zero.

For any subset $S \subseteq V$, let $\Gamma(S)$ denote the *cut set* in H defined by S, i.e., the set of edges in E with one endpoint in S and one endpoint in $V - S$, and define the (*edge*) *magnification* $\mu(H)$ of H by

$$\mu(H) = \min_{0<|S|\leq|V|/2} \frac{|\Gamma(S)|}{|S|}.$$

Clearly, $0 < \mu(H) \leq d$. The following equivalence is immediate from the definition of conductance:

Proposition 2.12 *Let G be the underlying graph of an ergodic random walk on a graph H with maximum degree d and transition probabilities β/d between distinct adjacent states. Then the conductance of G is given by*

$$\Phi(G) = \beta\mu(H)/d. \qquad \square$$

In a typical family of random walks $\mathcal{MC}(x)$, the degree d is fairly small (i.e., bounded by a polynomial in $|x|$), so the rapid mixing criterion boils down to finding polynomial lower bounds on the magnification. This view simplifies the analysis of families of this kind. As the following example illustrates, however, such a bound on the magnification is of no significance when the degree is large.

Example 2.13 For $n \in \mathbb{N}^+$, define the graph $H(n)$ as follows. Let H_1, H_2 be two copies of the complete graph $K_{N/2}$, where $N = 2^n$. Then $H(n)$ consists of the

disjoint union of H_1 and H_2 together with a perfect matching between vertices of H_1 and vertices of H_2. Let $MC(n)$ be a random walk on $H(n)$ as above. Such a chain is clearly ergodic, and it is not hard to verify that $\mu(H(n)) = 1$, with the minimum value of $\Gamma(S)/|S|$ attained when S is the vertex set of one of the H_i. However, the degree of $H(n)$ is $N/2 = 2^{n-1}$, which precludes the possibility of rapid mixing regardless of the value of β. An obstacle to rapid convergence is presented by the edges between the vertex sets of H_1 and H_2, which constitute a constriction to flow in the underlying graph. □

Chapter 3

Direct Applications

We have seen in Chapter 2 that the Markov chain paradigm provides an elegant general approach to generation problems, and have developed some theoretical machinery for analysing the efficiency of the resulting algorithms. The purpose of this chapter is to demonstrate the utility of the approach by applying it to some concrete and non-trivial examples. We shall show how to generate various combinatorial structures by constructing suitable ergodic Markov chains having the structures as states and transitions corresponding to simple local perturbations of the structures. The rate of convergence will be investigated using the techniques of Chapter 2, and in particular the rapid mixing characterisation of Corollary 2.8. In each case, the detailed structure of the Markov chain will enable us to estimate the conductance of its underlying graph, and we develop a useful general strategy for doing this. The major results of the chapter are the existence of efficient approximation algorithms for two significant #P-complete counting problems.

3.1 Some simple examples

Before tackling some more substantial problems, let us first apply the techniques of Chapter 2 to construct natural Markov chain generators for a few very simple structures. The generation problems considered in this section are not particularly interesting from a computational point of view as a number of efficient exact methods exist for their solution. Moreover, the associated counting problems are completely trivial. However, our analysis will serve to illustrate what

is involved in practical applications of the characterisation of Chapter 2. It will also allow us to develop some additional technology which will play a central rôle in later proofs.

Consider first the relation B which associates with each natural number n the set $B(n) = \{0,1\}^n$ of bit vectors of length n. We proceeed to construct a family of Markov chains $\mathcal{MC}(n)$ which can be used as an almost uniform generator for B following the paradigm of Figure 2.1. The most natural process to look at here is one which moves around the state space $B(n)$ by flipping a single random bit on each transition. Thus we can view $\mathcal{MC}(n)$ as a random walk on the n-regular graph $H(n)$ with vertex set $B(n)$ and edge set

$$\{(u,v) \in B(n) \times B(n) : \mathcal{D}(u,v) = 1\},$$

where $\mathcal{D}$ denotes Hamming distance. Of course, $H(n)$ is just the n-dimensional hypercube. To avoid problems of periodicity, we add a self-loop probability of $1/2$ to each state (i.e., $\beta = 1/2$ in the terminology of Proposition 2.12); note that this also dispenses with the problem of negative eigenvalues as in Proposition 2.3. This gives us an ergodic time-reversible Markov chain with uniform stationary distribution.

Turning now to the question of efficiency, it is clear that conditions (mc1)–(mc3) of Section 2.3 hold in this case: we may select 0^n as initial state and simulate individual steps in $O(\log n)$ time on an OCM (see Proposition 1.3). The efficiency of the generation procedure is therefore governed by the rate of convergence of the chain. The results of Section 2.3 in turn imply that this depends on the magnification of $H(n)$. Fortunately, a suitable bound on this quantity is not too hard to come by:

Theorem 3.1 *The n-dimensional hypercube has magnification $\mu(H(n)) \geq 1$.*

Before proving Theorem 3.1, let us first confirm that the resulting generation procedure is efficient.

Corollary 3.2 *Simulation of the above family of Markov chains yields a f.p. almost uniform generator for the bit vector relation B.*

Proof By the preceding discussion, it is enough to check the rapid mixing condition (mc4). Theorem 3.1 and Proposition 2.12 imply that the conductance of the

underlying graph of $\mathcal{MC}(n)$ is bounded below by $1/2n$. Clearly, the minimum stationary probability satisfies $\lg \pi_{\min}^{(n)} = -n$. Since the conditions of Corollary 2.8 are satisfied, we conclude that the family of Markov chains is rapidly mixing as required. □

Remark Inspection of the proof of Corollary 2.8 reveals that the number of simulation steps performed by the generator on input $\langle n, \epsilon \rangle$ is $O\left(n^2(n + \lg \epsilon^{-1})\right)$. □

We turn now to the proof of Theorem 3.1, which illustrates a technique which will be employed throughout this chapter.

Proof of Theorem 3.1 Let $N = 2^n$ be the number of states of $\mathcal{MC}(n)$. Our argument hinges on the following observation. Suppose it is possible to specify a canonical simple path in $H(n)$ between each ordered pair of distinct states in such a way that no oriented edge of $H(n)$ is contained in more than bN of the paths. If S is any subset of states with $0 < |S| \leq N/2$, then the number of paths which cross the cut from S to $\overline{S}$ is clearly

$$|S|(N - |S|) \geq |S|N/2.$$

Hence for any such S the number of cut edges $|\Gamma(S)|$ is bounded below by $|S|N/2bN = |S|/2b$, whence

$$\mu(H(n)) = \min_S \frac{|\Gamma(S)|}{|S|} \geq \frac{1}{2b}. \tag{3.1}$$

The problem of bounding $\mu(H(n))$ below can therefore be reduced to one of defining a collection of canonical paths in $H(n)$ which are "sufficiently edge disjoint", as measured by the parameter b.

We now proceed to define a suitable set of paths. Let $u = (u_i)_{i=0}^{n-1}$ and $v = (v_i)_{i=0}^{n-1}$ be distinct elements of $B(n)$, and $i_1 < \cdots < i_l$ be the positions in which u and v differ. Then for $1 \leq j \leq l$, the jth edge of the canonical path from u to v corresponds to a transition in which the i_jth bit is flipped from u_{i_j} to v_{i_j}.

Consider now an arbitrary transition t of $\mathcal{MC}(n)$ (or, equivalently, an oriented edge of $H(n)$); our aim is to bound the number of paths which contain t. Suppose that t takes state $w = (w_i)$ to state $w' = (w'_i)$ by flipping the value of w_k,

and let $P(t)$ denote the set of paths containing t, viewed as ordered pairs of states. Rather than counting elements of $P(t)$ directly, we will set up an *injective* mapping from $P(t)$ into the state space $B(n)$; this will yield an upper bound on the ratio b appearing in (3.1).

The mapping $\sigma_t : P(t) \to B(n)$ is defined as follows: given an ordered pair $\langle u, v \rangle \in P(t)$, set $\sigma_t(u, v) = (s_i)$, where

$$s_i = \begin{cases} u_i, & 0 \leq i \leq k; \\ v_i, & k < i < n. \end{cases}$$

Thus $\sigma_t(u, v)$ agrees with u on the first $k+1$ bits and with v on the remainder. Note that we can express this definition more succinctly as $\sigma_t(u, v) = u \oplus v \oplus w'$, where $\oplus$ denotes bitwise exclusive-or.

We claim that $\sigma_t(u, v)$ is an unambiguous *encoding* of the endpoints u and v, so that σ_t is indeed injective. To see this, simply note that

$$u_i = \begin{cases} s_i, & 0 \leq i \leq k; \\ w_i, & k < i < n; \end{cases} \qquad v_i = \begin{cases} w'_i, & 0 \leq i \leq k; \\ s_i, & k < i < n. \end{cases}$$

Hence u and v may be recovered from knowledge of t and $\sigma_t(u, v)$, so σ_t is injective. It follows immediately that $|P(t)| \leq N$; in fact, since all vectors (s_i) in the range of σ_t satisfy $s_k = w_k$, we have the stronger result

$$|P(t)| \leq |B(n)|/2 = N/2.$$

Since t was chosen arbitrarily, the number of paths traversing *any* oriented edge cannot exceed $N/2$. Setting $b = 1/2$, inequality (3.1) now yields the desired bound on the magnification $\mu(H(n))$. □

Remark The bound of Theorem 3.1 is tight. To see this, let S be the subset of $B(n)$ consisting of all vectors with first bit 0 and note that $|\Gamma(S)|/|S| = 1$. Hence $\mu(H(n)) = 1$ for the n-dimensional hypercube. □

Some observations on the above proof are in order here. The idea of path counting is quite general and has been used before in the literature to investigate the connectivity properties of various graphs in other contexts (see e.g. [56], in which the magnification of the hypercube is also studied). As we shall see later, it may be adapted to yield bounds on the conductance of the underlying graph

of an arbitrary time-reversible chain. The novelty of the proof lies in the use of the injective mapping technique to bound the number of paths which traverse an edge. This is not actually necessary in this simple example as the paths could have been counted explicitly. The point is that in more complex cases the states of the chain will be solutions of a non-trivial relation, such as matchings in a graph, and we will have no useful information about their number — indeed, this is what we will ultimately be trying to compute. It is then crucial to be able to bound the maximum number of paths through any edge in terms of the number of states without explicit knowledge of these quantities. This is precisely what the injective mapping technique achieves.

Other simple Markov chains may be analysed in a similar fashion. Examples of rapidly mixing families include random walks on n-dimensional cubes of side d and a host of "card-shuffling" processes whose state space is the set of permutations of n objects and whose transitions correspond to some natural shuffling scheme (see, e.g., [1]). Here we content ourselves with two further examples to illustrate the generality of our approach before passing on to more interesting problems.

Example 3.3 We will show how to construct an efficient Markov chain generator for subsets of an n-set of cardinality m. Equivalently, let SUBS be the relation which associates with pairs of natural numbers n, m all bit vectors of length n containing precisely m 1's. For each pair n, m with $0 < m < n$ define the Markov chain $\mathcal{MC}(n, m)$ with state space $\text{SUBS}(n, m)$ and transitions as follows: randomly select a pair of positions, one of which contains a 0 and the other a 1, and interchange their values. Adding a self-loop probability of $1/2$ to each state, we may view $\mathcal{MC}(n, m)$ as an ergodic random walk on a graph $H(n, m)$ of degree $m(n - m)$.

To get a bound on $\mu(H(n, m))$, we proceed as in the proof of Theorem 3.1. For $u, v \in \text{SUBS}(n, m)$, define the canonical path from u to v as follows: let $i_1 < \cdots < i_l$ denote those positions i for which $u_i = 0 \neq v_i$, and $j_1 < \cdots < j_l$ those j for which $u_j = 1 \neq v_j$. Then the kth step of the path from u to v involves interchanging u_{i_k} and u_{j_k}. Now let t be a transition which takes state w to w' by interchanging the values $w_i = 0$ and $w_j = 1$, and let $P(t)$ be the set of paths containing t. For each pair $\langle u, v \rangle \in P(t)$ we define the encoding $\sigma_t(u, v) = u \oplus v \oplus w'$ as before. It is easy to check that σ_t is again injective,

and that its range is the subset of $\text{SUBS}(n,m)$ consisting of vectors (s_k) with $s_i = 0$ and $s_j = 1$. It follows that $|P(t)| \leq (m(n-m)/n(n-1))N$, where $N = |\text{SUBS}(n,m)|$ is the total number of states. Appealing to (3.1) with the appropriate value of b now yields

$$\mu(H(n,m)) \geq \frac{n(n-1)}{2m(n-m)}.$$

The conductance of $MC(n,m)$ is therefore at least $n(n-1)/4m^2(n-m)^2 \geq 2/n^2$, and we have rapid mixing. □

Our final example is a card-shuffling process based on random transpositions.

Example 3.4 For a natural number n, let S_n denote the set of permutations of the set $[n] = \{0,\ldots,n-1\}$. Consider a deck of n cards labelled with the elements of $[n]$, and identify $\rho = (\rho_0,\ldots,\rho_{n-1}) \in S_n$ with the ordering of the deck in which the ith card from the top is ρ_i. We define a Markov chain $MC(n)$ with state space S_n in which transitions are made by picking a pair of cards at random (without replacement) and interchanging them. As usual, we incorporate a self-loop probability of $1/2$ for each state.

Once again we have an ergodic random walk on a regular graph of degree $n(n-1)/2$ and we need to look at its magnification. A canonical path between permutations u, v of the deck can be described as follows: for successive values $k = 0,\ldots,n-1$, move the card v_k into position k (if it is not there already) by interchanging it with the current kth card. Consider now some transition t which interchanges the cards in positions j and k of the permutation w, with $j > k$, and as before let $P(t)$ be the set of paths containing t. The injective mapping $\sigma_t : P(t) \to S_n$ is not quite so obvious here. For $\langle u,v\rangle \in P(t)$, we refer to the positions of a given card in u, v, w respectively as its *initial*, *final* and *current* positions. The permutation $\sigma_t(u,v)$ is then defined as follows:

(i) place the cards $w_0,\ldots,w_{k-1}$ in their initial positions;

(ii) place the remaining $n-k$ cards in the vacant positions in the order in which they appear in the final permutation v.

Let us now check that σ_t is injective. Given t and $\sigma_t(u,v)$ we can uniquely recover v as follows: the final positions of cards $w_0,\ldots,w_{k-1}$ are the same as

their current positions, and the final order of the remaining cards may be read off from $\sigma_t(u,v)$. To recover u, note that the initial positions of $w_0,\ldots,w_{k-1}$ are just as in $\sigma_t(u,v)$; but these positions together determine, for every i, the current position of the card initially in position i, since each previous transition on the path involved moving one of $w_0,\ldots,w_{k-1}$ into its final position. Hence we can deduce the initial positions of *all* cards, and so recover u.

Thus σ_t is injective, and we have $|P(t)| \leq N$, where $N = |S_n|$ is the total number of states. It follows from (3.1) and Proposition 2.12 that the conductance of $MC(n)$ is $1/2n(n-1)$, so this family of card-shuffling processes is rapidly mixing. The number of simulation steps required to achieve tolerance ϵ is easily seen from the proof of Corollary 2.8 and an application of Stirling's approximation to be $O\big(n^4(n\lg n + \lg \epsilon^{-1})\big)$. □

The Markov chain families mentioned in this section all possess a highly symmetrical structure which makes them particularly easy to analyse. These and similar processes have also been studied using other methods such as coupling, stopping times and group representation theory: see [1], [3], [17] for a variety of examples. The time bounds obtained by these methods are generally rather tighter than ours and can often be shown to be optimal. However, the full power of our approach will become apparent in the sequel where it will permit the analysis of highly non-symmetric chains with only a little additional effort. Most significantly, such chains have seemingly not proved amenable to analysis by any of the other established methods.

3.2 Approximating the permanent

In this section we treat our first major example — the groundwork of Chapter 2 begins to bear fruit in the form of a significant and unexpected approximability result.

The *permanent* of an $n \times n$ matrix A with 0-1 entries a_{ij} is defined by

$$\mathrm{per}\,(A) = \sum_{\sigma} \prod_{i=0}^{n-1} a_{i\sigma(i)}\,,$$

where the sum is over all permutations σ of the set $[n]$. Evaluating $\mathrm{per}(A)$ is equivalent to counting perfect matchings (1-factors) in the bipartite graph

$G = (V_1, V_2, E)$, where $V_1 = \{x_0, \ldots, x_{n-1}\}$, $V_2 = \{y_0, \ldots, y_{n-1}\}$ and $(x_i, y_j) \in E$ iff $a_{ij} = 1$. The permanent function arises naturally in a number of fields, including algebra, combinatorial enumeration and the physical sciences, and has been an object of study by mathematicians since first appearing in 1812 in the work of Cauchy and Binet. We shall mention an application in statistical physics in the next section; for further background information see [43].

Despite considerable effort, and in contrast with the syntactically very similar determinant, no efficient procedure for computing the permanent is known. Convincing evidence for its inherent intractability was provided in the late 1970s by Valiant [58], who demonstrated that the problem of counting perfect matchings in a bipartite graph is #P-complete. By contrast, the corresponding construction problem is solvable in polynomial time for arbitrary graphs [20]. The perfect matchings relation therefore belongs to the class discussed in Section 1.5, and the question of approximability is pertinent.

Until recently, little tangible progress had been made in the area of approximation algorithms for the permanent. In 1985, however, Broder [12] proposed a Markov chain approach for almost uniformly generating perfect matchings, which in view of the relationships of Section 1.4 could be used to count them. In this section, we show for the first time that this method does indeed yield a f.p. randomised approximate counter for perfect matchings in a large class of graphs, including all graphs which are sufficiently dense. The existence of an efficient randomised approximation algorithm for the dense permanent is therefore established. The crucial step is to show that the appropriate family of Markov chains on matchings is rapidly mixing.

Let $G = (V_1, V_2, E)$ be a bipartite graph with $|V_1| = |V_2| = n$, and for $k \in \mathbb{N}$ let $M_k(G)$ denote the set of matchings of size k in G. We assume throughout that G has a perfect matching, i.e., that $M_n(G)$ is non-empty. We view elements of E as unordered pairs of vertices, and matchings in G as subsets of E. If $A, B \subseteq E$ and $e \in E$ then $A \oplus B$ denotes the symmetric difference of A and B, while $A + e$ and $A - e$ denote the sets $A \cup \{e\}$, $A \setminus \{e\}$ respectively.

Following Broder [12], we proceed to define a Markov chain $\mathcal{MC}_{\mathrm{pm}}(G)$ with state space $\mathcal{N} = M_n(G) \cup M_{n-1}(G)$. Note that $\mathcal{N}$ includes auxiliary states, namely "near-perfect" matchings in G, which will permit free movement of the process between perfect matchings. Transitions in the chain are specified as

follows: in any state $M \in \mathcal{N}$, choose an edge $e = (u, v) \in E$ uniformly at random, and then

(i) if $M \in M_n(G)$ and $e \in M$, move to state $M' = M - e$ (*Type 1 transition*);

(ii) if $M \in M_{n-1}(G)$ and u, v are unmatched in M, move to $M' = M + e$ (*Type 2 transition*);

(iii) if $M \in M_{n-1}(G)$, u is matched to w in M, and v is unmatched in M, move to $M' = (M + e) - (u, w)$ (*Type 0 transition*);

(iv) in all other cases, do nothing.

For the sake of convenience, we introduce an additional self-loop probability of $1/2$ for each state; i.e., with probability $1/2$ the process does not select a random edge as above but simply remains at M. $\mathcal{MC}_{\mathrm{pm}}(G)$ may be viewed as a random walk on an appropriate graph H of maximum degree n. It is not hard to see that H is connected, so $\mathcal{MC}_{\mathrm{pm}}(G)$ is ergodic and time-reversible with uniform stationary distribution.

We now consider using the algorithm of Figure 2.1 in conjunction with the family of chains $\mathcal{MC}_{\mathrm{pm}}(G)$ as an almost uniform generator for perfect matchings. Stepwise simulation of transitions is readily performed by an OCM in time $O(|E|)$, and we have already noted that the construction problem for perfect matchings can be solved in polynomial time. Hence conditions (mc1) and (mc2) hold. Condition (mc3), however, presents a problem, since G may in general contain many more near-perfect than perfect matchings. Let us call G *dense* if its minimum vertex degree is at least $n/2$. It is not hard to check (see below) that, if G is dense, $|M_n(G)|/|\mathcal{N}| \geq 1/n^2$, so that (mc3) holds. Remarkably, under this assumption it is also possible to prove condition (mc4), i.e., that the family of Markov chains is rapidly mixing. This is a consequence of the following theorem.

Theorem 3.5 *For dense bipartite graphs G, the conductance of the underlying graph of the Markov chain $\mathcal{MC}_{\mathrm{pm}}(G)$ is at least $1/12n^6$.*

Proof It is sufficient to show that the graph H defining the random walk performed by $MC_{\mathrm{pm}}(G)$ has magnification

$$\mu(H) \geq 1/6n^4\,, \tag{3.2}$$

for then the theorem follows from Proposition 2.12 with $\beta/d = 1/2|E| \geq 1/2n^2$. To show (3.2) we proceed as in the proofs of the previous section by defining a set of canonical paths in H. If no transition occurs in more than $b|\mathcal{N}|$ of these, (3.1) gives us a bound on the magnification.

We begin by specifying, for each $M \in \mathcal{N}$, canonical paths to and from a unique closest perfect matching $\overline{M} \in M_n(G)$ as follows, where u, v denote the unmatched vertices (if any) of M:

(i) if $M \in M_n(G)$ then $\overline{M} = M$ and the path is empty;

(ii) if $M \in M_{n-1}(G)$ and $(u,v) \in E$, then $\overline{M} = M + e$ and the path consists of a single Type 2 transition;

(iii) if $M \in M_{n-1}(G)$ and $(u,v) \notin E$, fix some $(u',v') \in M$ such that $(u,v'),(u',v) \in E$: note that at least one such edge must exist by the density assumption on G. Then $\overline{M} = (M - (u',v')) + (u,v') + (u',v)$, and we specify one of the two possible paths of length 2 from M to $\overline{M}$, involving a Type 0 transition followed by a Type 2 transition.

The canonical path from $\overline{M}$ to M consists of the same edges of H traversed in the opposite direction.

For future reference, we observe that no perfect matching is involved in too many canonical paths of the above form: for $M \in M_n(G)$, define the set

$$K(M) = \{M' \in \mathcal{N} : \overline{M'} = M\}\,.$$

Then, since each matching in $K(M)$ has at least $n-2$ edges in common with M, it is easy to see that $|K(M)| \leq n^2$. Note that the sets $K(M)$ partition $\mathcal{N}$. This implies that $|\mathcal{N}| \leq n^2|M_n(G)|$, thus verifying our earlier claim that the near-perfect matchings are not too numerous. It is also worth noting that this is the only point in the proof at which the bipartite structure of G is used: we shall have more to say about this later.

Next we define a canonical path in H between an ordered pair I, F of *perfect* matchings (refer to Figure 3.1(a)). To do this, we first assume a fixed ordering of all even cycles of G, and distinguish a ***start vertex*** in each cycle. Now consider the symmetric difference $I \oplus F$; we may write this as a sequence $C_1, \ldots, C_r$ of disjoint even cycles, each of length at least 4, where the indices respect the above ordering. The path from I to F involves ***unwinding*** each of the cycles $C_1, \ldots, C_r$ in turn in the following way. Suppose the cycle C_i has start vertex u_0 and consists of the sequence of distinct vertices $(u_0, v_0, u_1, v_1, \ldots, u_l, v_l)$, where $(u_j, v_j) \in I$ for $0 \leq j \leq l$ and the remaining edges are in F. Then the first step in the unwinding of C_i is a Type 1 transition which removes the edge (u_0, v_0). This is followed by a sequence of l Type 0 transitions, the jth of which replaces the edge (u_j, v_j) by (u_j, v_{j-1}). The unwinding is completed by a Type 2 transition which adds the edge (u_0, v_l).

The canonical path between any pair of matchings $I, F \in \mathcal{N}$ is now defined as the concatenation of three segments as follows:

initial segment:	follow the canonical path from I to $\overline{I}$;
main segment:	follow the canonical path from $\overline{I}$ to $\overline{F}$;
final segment:	follow the canonical path from $\overline{F}$ to F.

Now consider an arbitrary oriented edge of H, corresponding to a transition t in the Markov chain. We aim to establish an upper bound of the form $b|\mathcal{N}|$ on the number of canonical paths which contain this transition. Suppose first that t occurs in the initial segment of a path from I to F, where $I, F \in \mathcal{N}$. Then it is clear from the definition of initial segment that the perfect matching $\overline{I}$ is uniquely determined by t. But we have already seen that $|K(\overline{I})| \leq n^2$. Since $I \in K(\overline{I})$, the number of paths which contain t in their initial segment is thus at most $n^2|\mathcal{N}|$. A symmetrical argument shows that the number of paths containing t in their final segment is similarly bounded.

To handle the main segments of the paths, we make use of the injective mapping technique seen earlier. This will obviate the need for any explicit counting of structures in $\mathcal{N}$, which is crucial here. Let t be a transition from M to M', where $M, M' \in \mathcal{N}$ are distinct, and denote by $P(t)$ the set of ordered pairs $\langle I, F \rangle$ of *perfect* matchings such that t is contained in the canonical path from I to F. We proceed to define, for each pair $\langle I, F \rangle \in P(t)$, an encoding $\sigma_t(I, F) \in \mathcal{N}$ from which I and F can be uniquely reconstructed. The intention is that, if

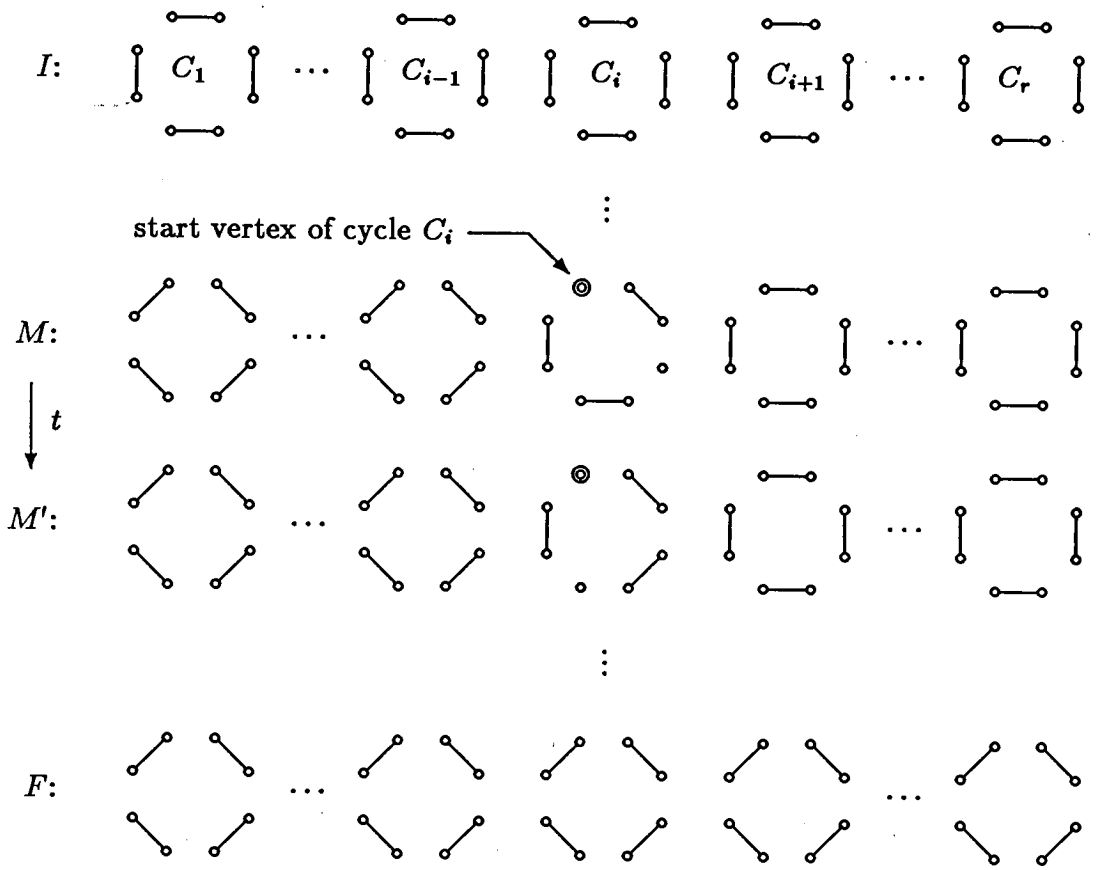

Figure 3.1(a): A transition t on the canonical path from I to F

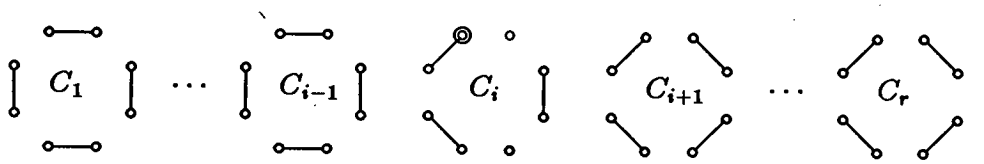

Figure 3.1(b): The corresponding encoding $\sigma_t(I, F)$

$C_1, \ldots, C_r$ is the ordered sequence of cycles in $I \oplus F$, and t is traversed during the unwinding of C_i, then the encoding should agree with I on $C_1, \ldots, C_{i-1}$ and on that portion of C_i which has already been unwound, and with F elsewhere.

With this in mind, consider the set $S = I \oplus F \oplus (M \cup M')$. Since $I \cap F \subseteq M \cup M' \subseteq I \cup F$ and $|I| = |F| = |M \cup M'| = n$, elementary set theory tells us that $|S| = n$. Furthermore, suppose that some vertex u has degree greater than 1 in S, i.e., we have $(u, v_1), (u, v_2) \in S$ for distinct vertices v_1, v_2. Then necessarily (u, v_1) and (u, v_2) both lie in $I \oplus F$, which in turn implies that neither edge lies in $M \cup M'$. Hence the vertex u must be unmatched in $M \cup M'$. From the form of the transitions, however, it is clear that $M \cup M'$ contains at most one such vertex $u = u_t$; moreover, this is the case iff t is a Type 0 transition, and u_t must then be the start vertex of the cycle currently being unwound. In this case, we denote by $e_{I,t}$ the edge of I incident with u_t.

We are now in a position to define the encoding:

$$\sigma_t(I, F) = \begin{cases} (I \oplus F \oplus (M \cup M')) - e_{I,t}, & \text{if } t \text{ is Type 0;} \\ I \oplus F \oplus (M \cup M'), & \text{otherwise.} \end{cases}$$

Figure 3.1(b) illustrates this definition for a Type 0 transition. In view of the above discussion, $\sigma_t(I, F)$ is always a matching of cardinality at least $n - 1$, and hence an element of $\mathcal{N}$. It remains for us to show that I and F can be recovered from it.

First observe that $I \oplus F$ can be recovered immediately using the relation

$$I \oplus F = \begin{cases} (\sigma_t(I, F) \oplus (M \cup M')) + e_{I,t}, & \text{if } t \text{ is Type 0;} \\ \sigma_t(I, F) \oplus (M \cup M'), & \text{otherwise.} \end{cases}$$

(Note that $e_{I,t}$ is the unique edge which must be added to $\sigma_t(I, F) \oplus (M \cup M')$ to ensure that $I \oplus F$ is a union of disjoint cycles.) Thus we may infer the ordered sequence $C_1, \ldots, C_r$ of cycles to be unwound on the path from I to F. The cycle C_i which is currently being unwound, together with its parity with respect to I and F, is then determined by the transition t. The parity of all remaining cycles may be deduced from M and the cycle ordering. Finally, the remaining portions of I and F may be recovered using the fact that $I \cap F = M \setminus (I \oplus F)$. Hence $\sigma_t(I, F)$ uniquely determines the pair $\langle I, F \rangle$, so σ_t is an injective mapping from $P(t)$ to $\mathcal{N}$.

The existence of σ_t ensures that $|P(t)| \leq |\mathcal{N}|$ for any transition t. Since also $|K(M)| \leq n^2$ for any perfect matching M, we see that t is contained in the main

segment of at most $n^4|\mathcal{N}|$ paths. Combining this with the results for initial and final segments derived earlier, we deduce that the maximum total number of paths which contain t is bounded by

$$\left(n^2 + n^2 + n^4\right)|\mathcal{N}| \leq 3n^4|\mathcal{N}|.$$

Taking $b = 3n^4$ in (3.1) yields (3.2), which completes the proof. □

The characterisation of Corollary 2.8 now ensures that the Markov chains $\mathcal{MC}_{\rm pm}(G)$ constitute a rapidly mixing family. (Note that the number of perfect matchings in G is at most $n!$, so the minimum stationary probability $\pi_{\rm min}^{(G)}$ of $\mathcal{MC}_{\rm pm}(G)$ satisfies $\lg \pi_{\rm min}^{(G)} \geq -cn \lg n$ for some constant c.) In the light of the discussion preceding Theorem 3.5 we therefore have

Corollary 3.6 *There exists a f.p. almost uniform generator for perfect matchings in dense bipartite graphs.* □

Remarks (a) In his original paper [12], Broder claimed that the above rapid mixing property holds under the same density assumption. His proof, which is based on coupling ideas, is complex and hard to penetrate. More seriously, as was first observed by Mihail [42], it also contains a fundamental error which apparently cannot be rectified: as a result, Broder has withdrawn his claim (see Erratum to [12]). We feel that this is compelling evidence of the unsuitability of coupling and related methods for the analysis of Markov chains which lack a high degree of symmetry.

(b) A chain with a rather better conductance bound is obtained by modifiying $\mathcal{MC}_{\rm pm}(G)$ slightly so that transitions are effected by selecting a random vertex in V_2 rather than a random edge. This gives us a random walk with transition probabilities $1/2n$ and the same bound on the magnification.

(c) In the case that G is the complete bipartite graph $K_{n,n}$, the Markov chain in (b) may be viewed as a scheme for shuffling a deck of $n+1$ cards in which the top card is repeatedly interchanged with another card selected at random. Of course, $\mathcal{MC}_{\rm pm}(K_{n,n})$ itself provides a generator for all permutations of n objects, albeit rather indirectly. By appropriate choice of G, we can also generate various natural restricted classes of permutations which satisfy the above density condition, such as displacements or ménage arrangements. □

Let us now return to the approximation of the permanent. Unfortunately, the introduction of the density assumption means that we are no longer working with a self-reducible relation, so we cannot apply the reduction of Theorem 1.14 to get an approximate counter for perfect matchings. However, the same result is achieved by a more specialised construction due to Broder [12]:

Corollary 3.7 *There exists a f.p. randomised approximate counter for perfect matchings in dense bipartite graphs, and hence a f.p. randomised approximation scheme for the permanent of dense square* 0-1 *matrices.*

To simplify the proof, we first derive an elementary statistical fact which says that, for a finite set S and a subset $U \subseteq S$, the ratio $|U|/|S|$ can be estimated efficiently by almost uniform sampling from S provided the ratio is not too small.

Proposition 3.8 *Let S be a finite set and U a subset of S. Write $p = |U|/|S|$. Suppose that t elements of S are selected independently and with replacement from an almost uniform distribution within tolerance $\xi \in (0,1]$, and let X denote the proportion of the sample which belong to U. Then for any $\delta \in (0,1]$, the sample size t required to ensure that*

$$\Pr\big(X \text{ approximates } p \text{ within ratio } 1+2\xi\big) > 1-\delta$$

is at most $(54/\xi^2 p)\ln(2/\delta)$.

Proof Writing p' for the expectation of X we have, since p' approximates p within ratio $1+\xi$,

$$\begin{aligned}
&\Pr\big(X \text{ approximates } p \text{ within ratio } 1+2\xi\big) \\
&\qquad \geq \Pr\big(X \text{ approximates } p' \text{ within ratio } 1+\xi/2\big) \\
&\qquad \geq \Pr\big(|X-p'| \leq \xi p'/3\big) \\
&\qquad \geq 1-2\exp(-\xi^2 p' t/27)\,,
\end{aligned}$$

where the last inequality is derived from Chernoff's bound on tails of the binomial distribution [6, Proposition 2.4]. This latter expression certainly exceeds $1-\delta$ provided $t \geq (54/\xi^2 p)\ln(2/\delta)$. □

Proof of Corollary 3.7 Let $G = (V_1, V_2, E)$ with $|V_1| = |V_2| = n$ be dense. For $l = 0, \ldots, n-1$, consider the graph G_l obtained from G by appending l vertices to each part of the bipartition, each new vertex being connected to all vertices of G in the opposite part. More precisely, G_l is the graph (V_1', V_2', E'), where $V_1' = V_1 \cup \{x_0, \ldots, x_{l-1}\}$, $V_2' = V_2 \cup \{y_0, \ldots, y_{l-1}\}$ and

$$E' = E \cup \{(u, y_i) : u \in V_1,\, i \in [l]\} \cup \{(u, x_i) : u \in V_2,\, i \in [l]\}.$$

Clearly G_l is dense. Now consider the set $\mathcal{N} = M_{n+l}(G_l) \cup M_{n+l-1}(G_l)$. By setting up explicit bijections, it is readily seen that

$$\begin{aligned} |M_{n+l}(G_l)| &= (l!)^2 |M_{n-l}(G)| \\ |M_{n+l-1}(G_l)| &= (l!)^2 \big(2l|M_{n-l}(G)| + |M_{n-l+1}(G)| \\ &\qquad + (l+1)^2 |M_{n-l-1}(G)|\big). \end{aligned} \tag{3.3}$$

This suggests a procedure for estimating the ratio $|M_{n-l}(G)|/|M_{n-l-1}(G)|$: define $p_1 = |M_{n-l}(G)|/|\mathcal{N}|$, $p_2 = |M_{n-l-1}(G)|/|\mathcal{N}|$. By simulating the Markov chain $\mathcal{MC}_{\mathrm{pm}}(G_l)$, generate almost uniformly, within small tolerance ξ, some number of elements of $\mathcal{N}$, and let s_1, s_2 be the proportions of the sample which correspond to matchings in G of size $n-l$ and $n-l-1$ respectively. Then $\dfrac{(l+1)^2 s_1}{(2l+1)s_2}$ is an estimator of the desired ratio p_1/p_2. Provided this is sufficiently accurate for each l, the product of the estimated ratios gives a good approximation to $|M_n(G)|$. More precisely, for any specified accuracy $\epsilon \in (0, 1]$ we can arrange for s_1, s_2 to approximate p_1, p_2 respectively within ratio $1 + \epsilon/4n$ with probability at least $1 - 1/8n$. Repeating this for each l, the final estimate approximates $|M_n(G)|$ within ratio $(1 + \epsilon/4n)^{2n} \leq 1 + \epsilon$ with probability $(1 - 1/8n)^{2n} \geq 3/4$.

To see that the necessary accuracy can be achieved in polynomial time, note first that

$$\frac{1}{n^2} \leq \frac{|M_k(G)|}{|M_{k+1}(G)|} \leq n^2 \qquad \text{for } 0 \leq k < n. \tag{3.4}$$

The lower bound is trivial, while the upper bound follows from the density assumption in the same manner as the bound on $|K(M)|$ in the proof of Theorem 3.5. Hence from (3.3) the proportion of matchings in $\mathcal{N}$ corresponding to $(n-l)$-matchings in G is at least

$$\frac{(l!)^2(2l+1)|M_{n-l}(G)|}{|\mathcal{N}|} \geq \frac{2l+1}{n^2((l+1)^2+1) + 2l + 1} \geq \frac{1}{3n^3}.$$

A similar bound holds for the proportion corresponding to $(n-l-1)$-matchings in G. Thus p_1 and p_2 are not too small. Putting $\xi = \epsilon/8n$ and $\delta = 1/8n$ in Proposition 3.8 now ensures that the sample size required to achieve the specified accuracy is polynomially bounded in n and ϵ^{-1}. □

The reader may be wondering at this stage whether the problem of counting perfect matchings remains difficult when restricted to dense graphs: if not, of course, the approximation results of this section would not be very exciting. The following result, which is proved in [12], serves to justify our approach.

Theorem 3.9 (Broder) *The problem of counting perfect matchings in dense bipartite graphs is #P-complete.* □

So far in this section we have concentrated exclusively on bipartite graphs because of their connection with the permanent. The Markov chain $\mathcal{MC}_{\mathrm{pm}}(G)$ can be applied without modification to arbitrary graphs G. In fact, the only point at which we have relied on the bipartite structure of G is in the definition of the sets $K(M)$ in the proof of Theorem 3.5 and the bound on their size. Let $G = (V, E)$ be an arbitrary graph with $|V| = 2n$. As before, we assume that G contains a perfect matching. Call G *dense* if its minimum vertex degree is at least n. This ensures that $K(M)$ for $M \in M_n(G)$ is still well defined, and that $|K(M)| \leq 2n^2$. The rest of the proof carries through as before, yielding $b = 8n^4$ and consequently $\mu(H) \geq 1/16n^4$. The conductance is therefore bounded below by $1/64n^6$. (This can again be improved if transitions are implemented by random vertex selection.) Since a construction analogous to that of Corollary 3.7 holds for general dense graphs, we have

Corollary 3.10 *There exists a f.p. almost uniform generator and a f.p. randomised approximate counter for perfect matchings in arbitrary dense graphs.* □

We conclude this section by examining the rôle played in our results by the density assumption. In the reduction of Corollary 3.7, we used it to prove the polynomial upper bound (3.4) on the ratios $|M_k(G)|/|M_{k+1}(G)|$. The proof of Theorem 3.5 makes use of an even stronger property of dense graphs, namely that $M_k(G)$ can be partitioned into classes of polynomially bounded size, one for

each element of $M_{k+1}(G)$, such that all matchings in a given class are "close to" the corresponding element of $M_{k+1}(G)$. In fact, it will turn out that everything works under the considerably weaker assumption that

$$\frac{|M_{n-1}(G)|}{|M_n(G)|} \leq q(n) \tag{3.5}$$

for some polynomial q, where $2n$ is the number of vertices in G.[1]

First we show that this condition implies a similar bound on *all* the ratios $|M_k(G)|/|M_{k+1}(G)|$, as in (3.4). This follows from the useful fact that the sequence $|M_k(G)|$ is *log-concave.*

Lemma 3.11 *For any graph G and positive integer k,*

$$|M_{k+1}(G)||M_{k-1}(G)| \leq |M_k(G)|^2.$$

Proof A proof of this fact using techniques from complex analysis can be found in [28, Theorem 7.1]. We present an elementary combinatorial proof which uses ideas seen elsewhere in this chapter.

We may assume that $|M_{k+1}(G)| > 0$, since the inequality is trivially true otherwise. Define the sets $A = M_{k+1}(G) \times M_{k-1}(G)$ and $B = M_k(G) \times M_k(G)$. Our aim is to show that $|A| \leq |B|$.

Note first that, for any two matchings M, M' in G, the symmetric difference $M \oplus M'$ consists of a set of disjoint simple paths (possibly closed) in G. Let us call such a path an *M-path* if it contains one more edge of M than of M'; an M'-path is defined similarly. Clearly, all other paths in $M \oplus M'$ contain equal numbers of edges from M and M'. Now for any pair $\langle M, M' \rangle \in A$, the number of M-paths in $M \oplus M'$ must exceed the number of M'-paths by precisely 2. We may therefore partition A into disjoint classes $\{A_r : 0 < r \leq k\}$, where

$$A_r = \{\langle M, M' \rangle \in A : M \oplus M' \text{ contains } r+1 \text{ } M\text{-paths and } r-1 \text{ } M'\text{-paths}\}.$$

Similarly, the sets $\{B_r : 0 \leq r \leq k\}$ with

$$B_r = \{\langle M, M' \rangle \in B : M \oplus M' \text{ contains } r \text{ } M\text{-paths and } r \text{ } M'\text{-paths}\}.$$

---

[1] By implication, any graph G satisfying (3.5) is assumed to have $|M_n(G)| > 0$.

partition B. The lemma will follow from the fact that $|A_r| \leq |B_r|$ for each $r > 0$. Let us call a pair $\langle L, L' \rangle \in B_r$ *reachable* from $\langle M, M' \rangle \in A_r$ iff $L \oplus L' = M \oplus M'$ and L is obtained from M by taking some M-path of $M \oplus M'$ and flipping the parity of all its edges with respect to M and M'. (This is analogous to unwinding the path in the manner of the proof of Theorem 3.5.) Clearly, the number of elements of B_r reachable from a given $\langle M, M' \rangle \in A_r$ is just the number of M-paths in $M \oplus M'$, namely $r+1$. Conversely, any given element of B_r is reachable from precisely r elements of A_r. Hence if $|A_r| > 0$ we have

$$\frac{|B_r|}{|A_r|} = \frac{r+1}{r} > 1,$$

which completes the proof of the lemma. □

Remark In [28], the tight inequality

$$|M_k(G)|^2 \geq \frac{(k+1)(m-k+1)}{k(m-k)} |M_{k+1}(G)||M_{k-1}(G)|$$

is proved, where $m = \lceil n/2 \rceil$ and n is the number of vertices in G. The bound in our proof can also be improved a little, but we will not labour this point here as simple log-concavity is quite adequate for our purposes. □

Next we show that (3.5) is sufficient to ensure rapid mixing for the Markov chain $\mathcal{MC}_{\mathrm{pm}}(G)$.

Theorem 3.12 *For any graph $G = (V, E)$ with $|V| = 2n$ and $|M_n(G)| > 0$, the conductance of the underlying graph of $\mathcal{MC}_{\mathrm{pm}}(G)$ is bounded below by*

$$\frac{1}{16|E|} \left(\frac{|M_n(G)|}{|M_{n-1}(G)|} \right)^2.$$

Proof The proof is similar to that of Theorem 3.5 with one or two additional technicalities. As before, let H be the graph defining the random walk performed by $\mathcal{MC}_{\mathrm{pm}}(G)$ and $\mathcal{N}$ its state space. We will need a variant of the canonical path counting argument which deals only with certain types of paths. For a subset S of states with $|S| \leq |\mathcal{N}|/2$, let $\overline{S} = \mathcal{N} \setminus S$ and define

$$\begin{aligned} S_1 &= S \cap M_n(G); & S_1' &= \overline{S} \cap M_n(G); \\ S_2 &= S \cap M_{n-1}(G); & S_2' &= \overline{S} \cap M_{n-1}(G). \end{aligned}$$

We claim that, when counting paths crossing the cut from S to $\overline{S}$, it is enough to consider paths from S_1 to S_2' and from S_2 to S_1'. To make this precise, write $r = |M_n(G)|/|M_{n-1}(G)|$ and $\alpha = |S_1|/|S_2|$, and let $p_1 = |S_1||S_2'|$ and $p_2 = |S_2||S_1'|$ be the numbers of paths from S_1 to S_2' and from S_2 to S_1' respectively. Clearly, p_1 and p_2 are respectively increasing and decreasing functions of α (for fixed r, $|S|$, $|\mathcal{N}|$); moreover, it is easy to check that they are equal when $\alpha = r$. Hence $\max\{p_1, p_2\}$ is minimised when $\alpha = r$, and the minimum value is

$$\frac{r|S|(|\mathcal{N}| - |S|)}{(1+r)^2} \geq \frac{r|S||\mathcal{N}|}{8},$$

since clearly $r \leq 1$.

Now all we need do is define canonical paths in H between elements of $M_n(G)$ (perfect matchings) and elements of $M_{n-1}(G)$ (near-perfect matchings). If no more than $b|\mathcal{N}|$ of these paths use any oriented edge of H, by analogy with (3.1) we will get the bound

$$\mu(H) \geq \frac{1}{8b}\left(\frac{|M_n(G)|}{|M_{n-1}(G)|}\right). \tag{3.6}$$

Henceforth we consider only paths from perfect to near-perfect matchings, the complementary case being symmetrical. As in the proof of Theorem 3.5, the canonical path from $I \in M_n(G)$ to $F \in M_{n-1}(G)$ is determined by the symmetric difference $I \oplus F$, which now consists of an ordered sequence $C_1, \ldots, C_r$ of disjoint cycles together with a single open path Q both of whose endpoints are matched in I but not in F. The canonical path from I to F proceeds by unwinding first the C_i as before and then Q in the obvious way, with one endpoint nominated as start vertex.

Now let t be a transition in $\mathcal{MC}_{\mathrm{pm}}(G)$ from M to M' and $P(t) \subseteq M_n(G) \times M_{n-1}(G)$ the set of canonical paths which contain t. For $\langle I, F\rangle \in P(t)$, we define the encoding $\sigma_t(I, F)$ exactly as in the proof of Theorem 3.5. A moment's reflection should convince the reader that $\sigma_t(I, F) \in M_{n-1}(G) \cup M_{n-2}(G)$, since we now get an additional pair of unmatched vertices arising from the open path Q. Note that this takes us outside the state space, but Lemma 3.11 will take care of this. Recovery of I and F from t and $\sigma_t(I, F)$ works essentially as before.

Hence σ_t is an injective mapping and we have, using Lemma 3.11,

$$\begin{aligned} |P(t)| &\leq |M_{n-1}(G)| + |M_{n-2}(G)| \\ &\leq |M_{n-1}(G)| \left(1 + \frac{|M_{n-1}(G)|}{|M_n(G)|}\right) \\ &= \frac{|M_{n-1}(G)|}{|M_n(G)|} |\mathcal{N}|. \end{aligned}$$

Thus we may take $b = |M_{n-1}(G)|/|M_n(G)|$ in (3.6) to get

$$\mu(H) \geq \frac{1}{8}\left(\frac{|M_n(G)|}{|M_{n-1}(G)|}\right)^2.$$

The final result is obtained via Proposition 2.12, the transition probabilities in $\mathcal{MC}_{\rm pm}(G)$ being $1/2|E|$. (As before, this can be improved slightly using a more intelligent implementation.) □

Corollary 3.13 *There exists a f.p. almost uniform generator and a f.p. randomised approximate counter for perfect matchings in any family of graphs satisfying* (3.5).

Proof The generator is immediate from the foregoing theorem. The counter follows via the reduction of Corollary 3.7, once we have noted from (3.3) that the graphs G_l inherit from G a bound of the form (3.5). □

Remark As observed by Dagum *et al* [16], the reduction from counting to generation described in Corollary 3.7 may be replaced by the following mehcanism, with a small increase in efficiency. In analogous fashion to $\mathcal{MC}_{\rm pm}(G)$, we may define for $1 \leq k \leq n$ a Markov chain $\mathcal{MC}_k(G)$ whose states are k- and $(k-1)$-matchings in G. (Thus $\mathcal{MC}_n(G)$ is just $\mathcal{MC}_{\rm pm}(G)$.) By a simple extension of the proof of Theorem 3.12, whereby multiple rather than unique canonical paths between states are counted, it can be shown that each of the chains $\mathcal{MC}_k(G)$ is rapidly mixing under the same condition (3.5) on G. This allows the ratios $|M_k(G)|/|M_{k-1}(G)|$ to be estimated directly for each k in turn. We do not dwell on this point here as we will present a more natural algorithm for the permanent in the next section. □

Our earlier results for dense graphs are now seen to be a special case of Corollary 3.13.[2] We might well want to ask whether other natural classes of graphs exist which satisfy (3.5) and for which membership of a given graph in the class can be tested easily. One observation which is sometimes useful in this connection is the following. An *augmenting path* for a matching $M \in M_{n-1}(G)$ is a sequence of transitions in the Markov chain $\mathcal{MC}_{\mathrm{pm}}(G)$ which takes M to a perfect matching. Suppose that G has maximum vertex degree d and that any $M \in M_{n-1}(G)$ has an augmenting path of length at most $l+1$. Then the ratio $|M_{n-1}(G)|/|M_n(G)|$ is clearly bounded above by nd^l. (Note that this is essentially the mechanism we used for dense graphs, with $l = 1$.) Thus G satisfies (3.5) for some fixed polynomial $q(n)$ if $l \leq$ constant $\times \log_d n$. This observation enables one to show that almost all graphs in fact satisfy (3.5). The following strong statement of this fact is due to Mark Jerrum:

Theorem 3.14 (Jerrum) *Let $G = (V_1, V_2, E)$ be a random bipartite graph with $|V_1| = |V_2| = n$, where each edge is selected independently with probability $p \geq (360 \log n)/n$. Then, with probability $1 - \mathrm{O}(n^{-1})$, G satisfies*

$$|M_{n-1}(G)| \leq n^{4+o(1)}|M_n(G)|\,. \qquad \square$$

The crux of the proof is that a.e. such graph has maximum degree $\mathrm{O}(pn)$ and augmenting paths of length $\mathrm{O}(\log_{pn} n)$. A direct consequence of Theorem 3.14 is the following:

Corollary 3.15 *There exists a f.p. almost uniform generator and a f.p. randomised approximate counter for perfect matchings in a.e. random bipartite graph as in Theorem* 3.14. $\square$

Analogous results hold in the non-bipartite case. Note that the bound on the density p is, up to a constant factor, equal to the threshold value for the existence of a perfect matching in such graphs, and hence essentially best possible.

---

[2] As Broder observes, the density bound quoted is tight in the sense that it is possible to construct, for any fixed $\delta > 0$, a sequence of graphs (G_n) with $2n$ vertices and minimum vertex degree $\geq n/(1+\delta)$ such that the ratio $|M_{n-1}(G_n)|/|M_n(G_n)|$ is exponentially large.

Other classes can often be tackled in similar fashion. For example, analogous results hold for a.e. random d-regular graph for any fixed d. Recently, Dagum *et al* [16] have also shown the existence of augmenting paths of constant length for all cn-regular bipartite graphs for any fixed $c > 0$.

This concludes our discussion of perfect matchings for now. An alternative view of these results will emerge as a by-product of our work on a different problem in the next section.

3.3 Monomer-dimer systems

This section is concerned with counting and generating all matchings (independent sets of edges) in a graph. Apart from their inherent interest, these problems arise in the theory of statistical physics, which is a rich source of combinatorial counting and generation problems.

A *monomer-dimer system* consists of a graph $G = (V, E)$, which is usually some form of regular lattice, whose vertices represent physical sites; adjacent pairs of sites may be occupied by diatomic molecules, or *dimers*. Configurations of the system correspond to arrangements of dimers on the lattice in which no two dimers overlap. In a configuration consisting of $k \leq |V|/2$ dimers, unoccupied sites are referred to as *monomers*, and the ratio $(|V| - 2k)/|V|$ is the *monomer density*. Monomer-dimer systems have been extensively studied as models of physical systems involving diatomic molecules. In the two-dimensional case, the system models the adsorption of dimers on the surface of a crystal. Three-dimensional systems occur in the theory of mixtures of molecules of different sizes and in the cell-cluster theory of the liquid state. For further information, see [28] and the references therein.

Most thermodynamic properties of such a system can be deduced from knowledge of the number of possible configurations, which is just the number of matchings in G. More generally, each edge e of G has an associated weight $c(e) \in \mathbb{R}^+$ which represents the relative probability of occupation by a dimer. This will depend on the contribution of such a dimer to the global energy of the system. The quantity of interest is the *partition function*

$$Z(G) = \sum_{M \in \text{MATCH}(G)} W(G, M), \tag{3.7}$$

where MATCH(G) is the set of matchings in G and $W(G,M) = \prod_{e\in M} c(e)$ is the *weight* of M. Counting matchings, i.e., the special case of (3.7) in which all edge weights are unity, is a #P-complete problem even when restricted to planar graphs [31], [59]. The main result of this section is that the more general sum (3.7) can in fact be approximated efficiently for any weighted graph G.

We shall again proceed via a related generation problem for matchings. Since the sum in (3.7) is weighted, however, matchings should be generated not uniformly but with probabilities proportional to their weights. In fact, sampling monomer-dimer configurations from the weighted distribution is an interesting problem in its own right as a means of estimating the expectation of various physical operators. We begin by generalising the concepts of Chapter 1 a little to handle weighted structures.

Let R be a relation over Σ, and $W : \Sigma^* \times \Sigma^* \to \mathbb{R}$ an arbitrary function. For any problem instance $x \in \Sigma^*$ and solution $y \in R(x)$, we call $W(x,y)$ the *weight* of y. We shall always assume that the weights are simple in the sense that W is computable in polynomial time. The weighted sum $\sum_{y\in R(x)} W(x,y)$ will be denoted by $\#_W R(x)$. The concept of self-reducibility can be extended to the weighted case as follows. We call the *pair* $\langle R,W\rangle$ *self-reducible* if R satisfies conditions (sr1)–(sr3) of Section 1.1 and in addition (with the same notation)

(sr4) There exists a polynomial time computable function $g : \Sigma^* \times \Sigma^* \times \mathbb{R} \to \mathbb{R}$ such that, for all $x \in \Sigma^*$,

$$\#_W R(x) = \sum_{w\in\Sigma^{\sigma(x)}} g\big(x, w, \#_W R(\psi(x,w))\big)$$

This definition merely generalises the idea that $\#R(x)$ can be computed easily given the values of $\#R$ for a few smaller problem instances. The *weighted counting problem* for the pair $\langle R,W\rangle$ involves computing the function $\#_W R$. By analogy with the unweighted case, this problem will be regarded as tractable if there exists a f.p. randomised approximation scheme for $\#_W R$, as defined in Section 1.2.

The monomer-dimer partition function (3.7) can be expressed as the weighted counting problem for a self-reducible pair as follows. Let MATCH be the relation which associates with a (weighted) graph G all matchings of G; this relation was seen to be self-reducible in Example 1.1. Let $c(e)$ denote the weight of the edge e

in G, and for $M \in \text{MATCH}(G)$ define $W(G,M) = \prod_{e \in E} c(e)$ as in (3.7). Then the pair $\langle \text{MATCH}, W \rangle$ is self-reducible since

$$\#_W\text{MATCH}(G) = \#_W\text{MATCH}(G^-) + c(e)\,\#_W\text{MATCH}(G^+),$$

where e is any edge of G and G^-, G^+ are as in Example 1.1.

In the case that all solution weights are positive, a corresponding *weighted generation problem* for $\langle R, W \rangle$ may be defined: this requires that each solution $y \in R(x)$ be output with probability proportional to its weight $W(x,y)$. The various definitions of generators given in Section 1.2 carry over in an obvious way to the non-uniform case. In particular, our main notion of tractability is an *almost W-generator* $\mathcal{G}$ for R, which is defined exactly as for an almost uniform generator except that, for all inputs $\langle x, \epsilon \rangle$ and solutions $y \in R(x)$, the output probabilities must satisfy

$$(1+\epsilon)^{-1}\phi(x,\epsilon)W(x,y) \leq \Pr\big(\mathcal{G}(x,\epsilon) = y\big) \leq (1+\epsilon)\phi(x,\epsilon)W(x,y)$$

for some function $\phi : \Sigma^* \times \mathbb{R}^+ \to (0,1]$. As before, $\mathcal{G}$ is fully-polynomial iff its runtime is bounded by a polynomial in $|x|$ and $\lg \epsilon^{-1}$.

The results of Section 1.3 transfer directly to the weighted case: thus we can work freely in the OCM model when considering approximation algorithms. Furthermore, it is easy to verify that, with minor modifications, the reductions of Theorems 1.10 and 1.14 also hold for weighted problems, giving the following generalisation of Corollary 1.15:

Corollary 3.16 *Let $R \subseteq \Sigma^* \times \Sigma^*$ and $W : \Sigma^* \times \Sigma^* \to \mathbb{R}^+$. If the pair $\langle R, W \rangle$ is self-reducible, then the following are equivalent:*

(i) *There exists a f.p. almost W-generator for R.*

(ii) *There exists a f.p. randomised approximation scheme for $\#_W R$.* □

Hence we will get a f.p. randomised approximation scheme for the monomer-dimer partition function provided we can generate matchings with probabilities roughly proportional to their weights. This we achieve using a suitable Markov chain simulation.

Given a graph $G = (V,E)$ with positive edge weights $\{c(e) : e \in E\}$, we consider the Markov chain $\mathcal{MC}_{\text{md}}(G)$ with state space $\mathcal{N} = \text{MATCH}(G)$ and transitions as follows: in any state $M \in \mathcal{N}$, choose an edge $e = (u,v) \in E$ uniformly at random and then

(i) if $e \in M$, move to $M - e$ with probability $1/(1 + c(e))$ (*Type 1 transition*);

(ii) if u, v are both unmatched in M, move to $M + e$ with probability $c(e)/(1 + c(e))$ (*Type 2 transition*);

(iii) if $e' = (u, w) \in M$ for some w, and v is unmatched in M, move to $(M + e) - e'$ with probability $c(e)/(c(e) + c(e'))$ (*Type 0 transition*);

(iv) in all other cases, do nothing.

As always, we simplify matters by adding a self-loop probability of $1/2$ to each state. It is then readily checked that $\mathcal{MC}_{\rm md}(G)$ is irreducible and aperiodic, and hence ergodic. The stationary probability π_M of $M \in \text{MATCH}(G)$ is easily seen to be proportional to its weight $W(G, M) = \prod_{e \in M} c(e)$, so the simulation procedure of Figure 2.1 will yield a f.p. almost W-generator for MATCH provided $\mathcal{MC}_{\rm md}(G)$ is rapidly mixing. Now $\mathcal{MC}_{\rm md}(G)$ is clearly time-reversible by virtue of the detailed balance condition (tr1), so we may again apply the analysis of Chapter 2. The crucial fact is the following:

Theorem 3.17 *For a graph $G = (V, E)$ with positive edge weights $\{c(e) : e \in E\}$, the conductance of the underlying graph of the Markov chain $\mathcal{MC}_{\rm md}(G)$ is bounded below by $1/(8|E|c_{\max}^2)$, where $c_{\max} = \max\{1, \max_{e \in E} c(e)\}$.*

Proof Let H be the underlying graph of $\mathcal{MC}_{\rm md}(G)$. The first step is to establish a weighted version of the path counting argument which led to the bound (3.1). Suppose that between each ordered pair $\langle I, F \rangle$ of distinct states we have a canonical path in H consisting only of edges of non-zero weight (corresponding to valid transitions in the chain). Furthermore, let us associate a weight $\pi_I \pi_F$ with the *path* from I to F. If S is any non-empty subset of states with capacity $C_S = \sum_{M \in S} \pi_M \leq 1/2$, the aggregated weight of all paths crossing the cut from S to $\overline{S} = \mathcal{N} \setminus S$ satisfies

$$\sum_{\substack{I \in S \\ F \in \overline{S}}} \pi_I \pi_F = C_S C_{\overline{S}} \geq C_S/2 \,. \tag{3.8}$$

Now let t be a transition from a state M to a state $M' \neq M$, and as usual denote by $P(t)$ the set of all ordered pairs $\langle I, F \rangle$ whose canonical path contains t.

Suppose it is known that, for any such transition t, the aggregated weight of paths containing t satisfies

$$\sum_{\langle I,F\rangle \in P(t)} \pi_I \pi_F \leq b w_t \,, \tag{3.9}$$

where $w_t = \pi_M p_{MM'} = \pi_{M'} p_{M'M}$ is the weight of the edge in H corresponding to t. ($p_{MM'}$ is the transition probability from M to M'.) Taking (3.8) and (3.9) together, we have the following bound on the ergodic flow out of S, where $\mathrm{cut}(S)$ denotes the set of transitions crossing the cut from S to $\overline{S}$:

$$\begin{aligned} F_S = \sum_{t \in \mathrm{cut}(S)} w_t &\geq b^{-1} \sum_{t \in \mathrm{cut}(S)} \sum_{\langle I,F\rangle \in P(t)} \pi_I \pi_F \\ &\geq b^{-1} \sum_{\substack{I \in S \\ F \in \overline{S}}} \pi_I \pi_F \\ &\geq \frac{C_S}{2b} \,. \end{aligned}$$

By definition, the conductance of H therefore satisfies

$$\Phi(H) \geq \frac{1}{2b} \,. \tag{3.10}$$

Our aim is thus to define a set of paths obeying a suitable bound b in (3.9).

To do this we generalise the proof of Theorem 3.5. Suppose there is an underlying order on all simple paths in G and designate in each of them a start vertex, which must be an endpoint if the path is not a cycle but is arbitrary otherwise. For distinct $I, F \in \mathcal{N}$, we can write the symmetric difference $I \oplus F$ as a sequence $S_1, \ldots, S_r$ of disjoint paths which respects the ordering. The canonical path from I to F involves unwinding each of the S_i in turn as follows. There are two cases to consider:

(i) *S_i is not a cycle.* Let S_i consist of the sequence $(v_0, v_1, \ldots, v_l)$ of vertices, with v_0 the start vertex. If $(v_0, v_1) \in F$, perform a sequence of Type 0 transitions replacing (v_{2j+1}, v_{2j+2}) by (v_{2j}, v_{2j+1}) for $j = 0, 1, \ldots$, and finish with a single Type 2 transition if l is odd. If on the other hand $(v_0, v_1) \in I$, begin with a Type 1 transition removing (v_0, v_1) and proceed as before for the reduced path $(v_1, \ldots, v_l)$.

(ii) *S_i is a cycle.* Let S_i consist of the sequence $(v_0, v_1, \ldots, v_{2l+1})$ of vertices, where $l \geq 1$ and $(v_{2j}, v_{2j+1}) \in I$ for $0 \leq j \leq l$, the remaining

edges belonging to F. Suppose that v_0 is the start vertex. Then the unwinding begins with a Type 1 transition to remove (v_0, v_1). We are left with a path with endpoints v_0, v_1, one of which must be the start vertex of the path. Suppose v_k, $k \in \{0,1\}$, is *not* the start vertex. Then we unwind the path as in (i) above but treating v_k as the start vertex. This trick serves to distinguish paths from cycles, as will prove convenient shortly.

Now let t be a transition from M to $M' \neq M$. The next step is to define our injective mapping $\sigma_t : P(t) \to \mathcal{N}$. As in the proof of Theorem 3.5, we set $\sigma_t(I,F)$ equal to $I \oplus F \oplus (M \cup M')$, and remove the edge $e_{I,t}$ of I adjacent to the start vertex of the path currently being unwound if necessary: this is so iff the path is a cycle and t is Type 0. It is now easily seen that $\sigma_t(I,F)$ consists of independent edges, and so is an element of $\mathcal{N}$. The difference $I \oplus F$ can be recovered from $\sigma_t(I,F)$ using the relation

$$I \oplus F = \begin{cases} (\sigma_t(I,F) \oplus (M \cup M')) + e_{I,t}, & \text{if } t \text{ is Type 0 and the current} \\ & \quad \text{path is a cycle;} \\ \sigma_t(I,F) \oplus (M \cup M'), & \text{otherwise.} \end{cases}$$

Note that we can tell whether the current path is a cycle from the sense of unwinding. Recovery of I and F themselves now follows as before from the path ordering. Hence σ_t is injective.

Moreover, it should be clear that $\sigma_t(I,F)$ is very nearly the complement of M in the union of I and F viewed as a multiset, so that the product $\pi_I \pi_F$ is approximately equal to $\pi_M \pi_{\sigma_t(I,F)}$, giving us a handle on b in (3.9). We now make this precise.

Claim *For any* $\langle I,F \rangle \in P(t)$, *we have*

$$\pi_I \pi_F \leq 4|E| c_{\max}^2 w_t \pi_{\sigma_t(I,F)} . \tag{3.11}$$

The Claim will be proved in a moment. First note that it immediately yields the desired bound b in (3.9), since for any transition t we have

$$\sum_{\langle I,F \rangle \in P(t)} \pi_I \pi_F \leq 4|E| c_{\max}^2 w_t \sum_{\langle I,F \rangle \in P(t)} \pi_{\sigma_t(I,F)} \leq 4|E| c_{\max}^2 w_t ,$$

where the second inequality follows from the fact that σ_t is injective. We may therefore take $b = 4|E| c_{\max}^2$, which in the light of (3.10) gives the conductance bound stated in the theorem.

It remains only for us to prove the Claim. We distinguish three cases:

(i) *t is a Type 1 transition.* Suppose $M' = M - e$. Then $\sigma_t(I,F) = I \oplus F \oplus M$, so, viewed as multisets, $M \cup \sigma_t(I,F)$ and $I \cup F$ are equal. Hence we have

$$\begin{aligned} \pi_I \pi_F &= \pi_M \pi_{\sigma_t(I,F)} \\ &= (w_t / p_{MM'}) \pi_{\sigma_t(I,F)} \\ &= 2|E|(1 + c(e)) w_t \pi_{\sigma_t(I,F)}, \end{aligned}$$

from which (3.11) follows.

(ii) *t is a Type 2 transition.* This is handled by a symmetrical argument to (i) above, with M replaced by M'.

(iii) *t is a Type 0 transition.* Suppose $M' = (M + e) - e'$, and consider the multiset $\sigma_t(I,F) \cup M$. This is equal to the multiset $I \cup F$ except that the edge e, and possibly also the edge $e_{I,t}$ are absent from it. Assuming $e_{I,t}$ is absent, which happens precisely when the current path is a cycle, we have

$$\begin{aligned} \pi_I \pi_F &= c(e_{I,t}) c(e) \pi_M \pi_{\sigma_t(I,F)} \\ &= c(e_{I,t}) c(e) (w_t / p_{MM'}) \pi_{\sigma_t(I,F)} \\ &= 2|E| c(e_{I,t}) (c(e) + c(e')) w_t \pi_{\sigma_t(I,F)}, \end{aligned}$$

which again satisfies (3.11). If $e_{I,t}$ is not absent, the argument is identical with the factor $c(e_{I,t})$ omitted.

This concludes the proof of the Claim and the theorem. □

Corollary 3.18 *There exists a f.p. almost W-generator for matchings in arbitrary weighted graphs, where W is the weighting function for matchings defined above, provided the edge weights are positive and presented in unary.*

Proof Define $c_{\min} = \min\{1, \min_{e \in E} c(e)\}$. Then the minimum stationary state probability in $MC_{\mathrm{md}}(G)$ is at least $c_{\min}^n/(2^{|E|}c_{\max}^n)$, where $n = |V|$. The logarithm of this quantity is at least $-p(|x|)$, where $|x|$ is the size of the input description and p is a polynomial. Hence by Theorem 3.17 and Corollary 2.8 the Markov chain family is rapidly mixing. Simulation of $MC_{\mathrm{md}}(G)$ by an OCM is a simple matter, and we may take the empty matching as initial state. □

In view of Corollary 3.16, we may now state the main result of this section:

Corollary 3.19 *There exists a f.p. randomised approximation scheme for the monomer-dimer partition function of arbitrary weighted graphs with edge weights presented in unary.* □

As we have already mentioned, the monomer-dimer chain provides some further insight into the results of the previous section. In particular, it yields alternative and arguably more natural algorithms for generating and counting *perfect* matchings in families of graphs satisfying (3.5), and thus a new proof of Corollary 3.13. The key to these algorithms is the introduction of carefully chosen edge weights.

Let $\mathcal{F}$ be a family of graphs satisfying (3.5) for some fixed polynomial q, and $G = (V, E)$ be a member of $\mathcal{F}$ with $|V| = 2n$. We introduce the notation $G(c)$ to stand for the graph obtained by giving each edge of G the weight c. To generate perfect matchings in G almost uniformly, we proceed as follows: for some suitable $c > 1$, use the Markov chain $MC_{\mathrm{md}}(G(c))$ as above to generate matchings in $G(c)$ (or equivalently in G) from the distribution determined by the edge weights, and fail if the ouptut is not a perfect matching. For any value of c, the induced distribution on perfect matchings is clearly uniform. Now suppose we choose $c = 2q(n)$; since all edge weights are polynomially bounded, Corollary 3.18 ensures that the generator is f.p. It remains only to check that the failure probability is not too large.

Writing m_k in place of $|M_k(G)|$, the log-concavity of the m_k (Lemma 3.11) implies that

$$\frac{m_k}{m_n} = \prod_{i=k}^{n-1} \frac{m_i}{m_{i+1}} \leq q(n)^{n-k}$$

for $0 \leq k \leq n$. In the stationary distribution of $MC_{\mathrm{md}}(G(c))$, the probability of being at a perfect matching is therefore

$$\frac{m_n(2q(n))^n}{\sum_{k=0}^{n} m_k(2q(n))^k} \geq \frac{2^n}{\sum_{k=0}^{n} 2^k} > \frac{1}{2}.$$

This confirms that the method works. Notice how the condition (3.5) again arises naturally here: in the absence of a polynomial bound on the ratio m_{n-1}/m_n, very large edge weights would be required to ensure that perfect matchings appear frequently enough.

Now let us review the problem of *counting* perfect matchings. The monomer-dimer chain suggests a more natural reduction to generation than that of Corollary 3.7. For a graph $G = (V, E)$ as above, we will estimate the ratios m_{k+1}/m_k in turn in a sequence of n stages, for $k = 0, \ldots, n-1$; an approximation to m_n is then obtained as the product of the estimated ratios. The main idea is to compute a sequence $c_1, \ldots, c_{n-1}$ of edge weights with the property that, in the stationary distribution of the Markov chain $MC_{\mathrm{md}}(G(c_k))$, the probability of being at a k-matching is quite large, allowing the ratio m_{k+1}/m_k to be estimated statistically. That such a sequence of weights exists is a consequence of the log-concavity of the m_k. Ideally, we want the weight c_k to be the inverse ratio m_{k-1}/m_k; however, it will suffice to substitute the *estimate* of this quantity obtained in the previous stage.

Figure 3.2 shows the approximate counter for perfect matchings, where G is the input graph and $\epsilon \leq 1$ the relative error specified for the final approximation. $\mathcal{G}$ is the almost W-generator for matchings described earlier, i.e., the call $\mathcal{G}(G(c), \cdot)$ invokes a simulation of the Markov chain $MC_{\mathrm{md}}(G(c))$. Line (1) and the iterations of the for-loop in line (2) correspond to the n stages of the computation mentioned above. Let c_{k+1} be the value of the weight parameter c at the end of the kth stage: we claim that, for each k, c_{k+1} approximates m_k/m_{k+1} within ratio $1 + \epsilon/2n$ with high probability, provided the value t in line (4) is suitably chosen. This will imply that the product Π output in line (8) approximates m_n within ratio $(1 + \epsilon/2n)^{n-1} \leq 1 + \epsilon$ with high probability, as required.

The claim is proved by induction on k. The base case is trivial since, from line (1), $c_1 = |E|^{-1} = m_0/m_1$. Now assume inductively that c_k approximates m_{k-1}/m_k within ratio $1 + \epsilon/2n$. In the stationary distribution of the chain

(1) $c := |E|^{-1}$; $\Pi := |E|$;
(2) **for** $k := 1$ **to** $n-1$ **do begin**
(3) **if** $c > 2q(n)$ **then halt** with output 0
else begin
(4) make t calls of the form $\mathcal{G}(G(c), \epsilon/12n)$
and let Y be the set of outputs;
(5) $\tilde{p}_k := |Y \cap M_k(G)|/t$; $\tilde{p}_{k+1} := |Y \cap M_{k+1}(G)|/t$;
(6) **if** $\tilde{p}_{k+1} = 0$ **then halt** with output 0
(7) **else begin** $c := c\tilde{p}_k/\tilde{p}_{k+1}$; $\Pi := \Pi/c$ **end**
end;
(8) **halt** with output Π

Figure 3.2: Approximate counter for perfect matchings

$MC_{\mathrm{md}}(G(c_k))$, let p_i, $0 \le i \le n$, denote the probability of being at an i-matching. Then clearly $m_k/m_{k+1} = c_k p_k/p_{k+1}$. Inspection of line (7) reveals that $c_{k+1} = c_k\tilde{p}_k/\tilde{p}_{k+1}$, where $\tilde{p}_k, \tilde{p}_{k+1}$ are estimates of p_k, p_{k+1} computed in lines (4) and (5) by observing the proportions of k-matchings and $(k+1)$-matchings in a sample produced by the generator. Since the tolerance is $\xi = \epsilon/12n$, by making the sample size t sufficiently large we can ensure that these estimates are within ratio $1+2\xi = 1+\epsilon/6n$ with high probability. Then c_{k+1} approximates m_k/m_{k+1} within ratio $(1+\epsilon/6n)^2 \le 1+\epsilon/2n$ with high probability, completing the inductive step of the proof. Note that the pathological cases of lines (3) and (6) can occur only in the unlikely event that some estimate c_k is out of range.

Finally, we need to investigate the runtime of the procedure. With the exception of lines (4) and (5), this is evidently bounded by a polynomial in n. Moreover, the bound on the edge weights in line (3) ensures that each call to $\mathcal{G}$ is polynomially bounded in n and ϵ^{-1}. We just have to check that t need only be similarly bounded in order to give good estimates $\tilde{p}_k, \tilde{p}_{k+1}$ with high probability. To see this, assume that c_k is a good estimate of m_k/m_{k+1} and note that, for $i \ge k$,

$$\frac{p_k}{p_i} = \frac{m_k {c_k}^k}{m_i {c_k}^i} = {c_k}^{k-i} \prod_{j=k}^{i-1} \frac{m_j}{m_{j+1}}. \tag{3.12}$$

By log-concavity of the m_i, each ratio in the product is bounded below by m_{k-1}/m_k. Also, $c_k{}^{k-i}$ approximates $(m_k/m_{k-1})^{i-k}$ within ratio $(1+\epsilon/2n)^{i-k} \leq (1+1/n)^n \leq e$. It follows from (3.12) that $p_k/p_i \geq 1/e$ for $i \geq k$. Exactly the same bound holds for $i < k$. Since $\sum p_i = 1$, we conclude that $p_k \geq (en+1)^{-1}$ and so is bounded below by an inverse polynomial in n. Moreover, we have

$$\frac{p_{k+1}}{p_k} = \frac{c_k m_{k+1}}{m_k} \geq \frac{1}{2}\left(\frac{m_0}{m_1}\right)\left(\frac{m_n}{m_{n-1}}\right) \geq \frac{1}{2|E|q(n)}$$

using log-concavity and (3.5), so that p_{k+1} is similarly bounded below. These lower bounds imply by Proposition 3.8 (generalised to the weighted case) that the sample size t required to ensure that $\tilde{p}_k, \tilde{p}_{k+1}$ approximate p_k, p_{k+1} within ratio $1+\epsilon/6n$ with high probability is polynomially bounded in n and ϵ^{-1}. Hence the procedure of Figure 3.2 constitutes a f.p. randomised approximate counter for perfect matchings in families of graphs satisfying (3.5).

Recall from Theorem 3.14 of the previous section that almost all sufficiently dense graphs satisfy (3.5) for a fixed polynomial q. However, it is not at all clear how to decide whether a given graph satisfies the bound. The above techniques suggest a simple randomised procedure for doing this. More precisely, let q be an arbitrary polynomial and suppose we wish to design an efficient algorithm which, when presented with an arbitrary graph G containing a perfect matching, behaves as follows:

(i) if $m_{n-1}/m_n \leq q(n)$, the algorithm accepts with high probability;

(ii) if $m_{n-1}/m_n > 6q(n)$, the algorithm rejects with high probability.

For intermediate values of the ratio, we do not care whether the algorithm accepts or rejects. (The value 6 here is used for illustrative purposes only and can be replaced by any fixed constant greater than 1.) Such an algorithm is of significant practical value, since we can be almost certain that any graph accepted satisfies $m_{n-1}/m_n \leq 6q(n)$, allowing us to count and generate perfect matchings in it. Moreover, all graphs with $m_{n-1}/m_n \leq q(n)$ will almost always be accepted.

To obtain an algorithm with this behaviour, all we need do is generate some number t of matchings in G using the chain $MC_{\mathrm{md}}(G(2q(n)))$ with some small fixed tolerance ϵ, and note the proportion s of these which are perfect. We accept iff $s \geq 3/8$. As we have already seen, in the case $m_{n-1}/m_n \leq q(n)$ the expected value of s will be $> 1/2(1+\epsilon)$, so taking $t = A \ln \delta^{-1}$, for a suitable

constant A and any desired $\delta > 0$, ensures that $\Pr(s < 3/8) < \delta$. Similarly, if $m_{n-1}/m_n > 6q(n)$ the expected value of s is $< (1+\epsilon)2q(n)/8q(n) = (1+\epsilon)/4$, and $\Pr(s \geq 3/8)$ can again be made arbitrarily small as above.

3.4 Concluding remarks

As well as presenting several positive results, this chapter has left open a number of interesting questions which we now briefly mention.

The first question which suggests itself concerns the approximability of the unrestricted permanent. Though based on different Markov chains, both approximation algorithms we have employed here give rise naturally to condition (3.5), that the ratio of near-perfect to perfect matchings must be polynomially bounded. On the other hand, there seems to be no *a priori* reason why counting perfect matchings in graphs which violate this condition should be particularly hard. The task of devising an efficient universally valid approximate counter for perfect matchings, or alternatively showing that the general problem is hard in the manner of Theorem 1.17, seem to be thorny issues which may require other techniques for their resolution.

As mentioned earlier, it would be of interest to know whether other easily characterised families of graphs, apart from dense graphs, satisfy (3.5). For example, this question is important in the case of certain regular lattice graphs, in which physicists are keen to count *dimer coverings* (monomer-dimer configurations with zero monomer density). Much effort has been expended on this problem, and an elegant exact solution obtained for planar lattices (or indeed arbitrary planar graphs) [35]. The three-dimensional case, however, remains open. Some small-scale experimentation suggests that the ratio of near-perfect to perfect matchings may well be polynomially bounded for three-dimensional rectangular lattices, allowing the problem to be efficiently approximated, but we have not been able to confirm this analytically. (Note that the straightforward augmenting path approach is of no help for such graphs due to their large diameter.) Another quantity of interest is the number of monomer-dimer configurations with given monomer density, i.e., the number of matchings of given cardinality. Both algorithms we have presented for counting perfect matchings can obviously be adapted to compute $|M_k(G)|$ for any given k, but we are left

with a similar condition on G, namely that the ratio $|M_{k-1}(G)|/|M_k(G)|$ be polynomially bounded.

From a practical point of view, it would be interesting to know whether the conductance bounds we have derived can be significantly improved. We make no claim of optimality here, prefering to concentrate on giving a clear exposition of the rapid mixing property. The practical utility of our algorithms, however, is likely to depend on rather tighter bounds being available.

Undoubtedly the biggest outstanding question is whether the techniques of this chapter can be applied to other interesting combinatorial structures. In many cases, a Markov chain on the structures with the desired stationary distribution suggests itself naturally. We give a few examples as an illustration:

Example 3.20 For a connected graph G, consider the Markov chain $\mathcal{MC}(G)$ whose states are the spanning trees of G, with transitions as follows: given a spanning tree T, select an edge e of G uniformly at random. If e belongs to T, do nothing; otherwise, add e to T to form a graph with a single cycle, and remove a randomly chosen edge of the cycle. Then $\mathcal{MC}(G)$ is symmetric, so its stationary distribution is uniform over the spanning trees of G. (As noted in Chapter 1, efficient exact methods do exist for counting and generating spanning trees. However, the simplicity of this Markov chain, together with the fact that it has certain features in common with a number of other natural chains, make it an interesting object of study in its own right.) □

Example 3.21 Let n be a positive integer and $\mathbf{g} = (g_0, \ldots, g_{n-1})$ a degree sequence on $[n]$, i.e., a sequence of natural numbers g_i with $\sum g_i$ even. Let GRAPHS(g) denote the set of all simple graphs on vertex set $[n]$ in which vertex i has degree g_i, and consider the Markov chain $\mathcal{MC}(\mathbf{g})$ defined as follows. The states of $\mathcal{MC}(\mathbf{g})$ are the elements of GRAPHS(g), together with all simple graphs on $[n]$ in which one pair i, j of vertices have deficient degrees $g_i - 1$, $g_j - 1$ and each other vertex k has degree g_k. To make a transition from state G, select a pair i, j of vertices uniformly at random and

(i) if $G \in$ GRAPHS(g) and (i, j) is an edge of G, delete (i, j) from G;

(ii) if $G \notin$ GRAPHS(g), (i, j) is not an edge of G, and the degree of i in G is $g_i - 1$, add (i, j) to G and, if this causes the degree of j to exceed g_j, remove a randomly chosen edge of G incident with j.

Once again, the stationary distribution is uniform over GRAPHS(g). (We shall attack the generation problem for the relation GRAPHS by a completely different method in the next chapter.) □

Example 3.22 Let n be a positive integer and ω a partial order on the set $[n]$. Let LE be the relation which associates with the pair $\langle n, \omega \rangle$ the set of all linear extensions of ω. A Markov chain $MC(n, \omega)$ with state space $\mathrm{LE}(n, \omega)$ and uniform stationary distribution can be defined as follows: given a linear extension μ of ω, select a pair i, j of elements of $[n]$ uniformly at random and interchange them in μ provided the resulting order is still consistent with ω. □

Similar (usually non-uniform) Markov chains have been the subject of extensive experimental investigation in statistical physics: examples include self-avoiding walks [9] and Ising (spin-glass) configurations [10] on various lattices. Typically, non-rigorous *ad hoc* arguments are given to justify the number of simulation steps performed in order to approximate the desired distribution on the states. Conclusions drawn from such experiments therefore often demand from the reader a certain act of faith.

Since almost all natural chains of this kind are time-reversible, the characterisation of Chapter 2 can in principle be applied to obtain rigorous bounds on the number of simulation steps required to achieve any specified level of accuracy. We conjecture that this is possible for other chains, and that the path counting technique developed in this chapter is a promising approach for obtaining upper bounds. Initial investigations with other structures suggest that it may need some refinement, however.

Finally, let us briefly mention the currently fashionable stochastic optimisation technique known as *simulated annealing* [37]. Here one weights system configurations according to the value of some cost function, and attempts to find a configuration of maximum weight by simulating a Markov process as above; the parameters of the system are gradually scaled so as to make maximum weight states absorbing. While asymptotic convergence can be proved (this question is non-trivial because the process is non-homogeneous), virtually nothing is known in precise terms about the rate of convergence. Since the question is essentially one of rapid mixing for an appropriate sequence of Markov chains, the methods described here potentially provide a means of analysis.

Chapter 4

Indirect Applications

In this chapter, we return to the general framework of Chapter 1 and re-examine the notions of approximate counting and almost uniform generation in the light of our work on Markov chains. Our main result is a dramatic improvement of the reduction from generation to counting for self-reducible relations presented in Theorem 1.10, which allows much larger errors in the counter to be handled. The reduction is achieved by constructing an ergodic Markov chain based on the tree of derivations. As always, the crucial feature of the chain from our point of view is that it converges rapidly to its stationary distribution. The machinery developed in Chapter 2 will enable us to establish this property painlessly.

This result has two major consequences. Firstly, it gives us a notion of approximate counting which is *robust* with respect to polynomial time computation in the following sense: for a self-reducible relation R, if randomised approximate counting within ratio $1 + \mathrm{O}(n^\beta)$ is possible in polynomial time for some real constant β, however large, then there exists a f.p. randomised approximate counter for R. In other words, a very crude counter can always be effectively bootstrapped to achieve arbitrarily good asymptotic behaviour. Secondly, it suggests a new practical technique for generating combinatorial structures when some asymptotic information about their number is available (to within a constant factor, say).

As a concrete example of this technique in action, we consider the problem of generating labelled graphs with given vertex degrees and a given excluded subgraph. Using a result of McKay [40] which provides analytic counting estimates for these graphs, we show that it is possible to generate them in poly-

nomial time from a distribution which is very close to uniform provided only that the maximum degree grows no faster than $O(m^{1/4})$, where m is the number of edges. In spite of the fact that the problem is apparently not self-reducible under this restriction, our technique can still be applied with a little extra work. This result represents a considerable improvement on hitherto known methods ([11], [55], [62]). It also implies the existence of polynomial time randomised algorithms for counting such graphs to within a factor of $1+m^{-\beta}$, for any real β.

4.1 A robust notion of approximate counting

The main aim of this section is to show how to construct a f.p. almost uniform generator for a self-reducible relation given only very approximate counting information for the structures concerned.

Let us first briefly review the straightforward reduction from generation to counting used to establish Theorem 1.10. This involves choosing a random path from the root of the tree of derivations to a leaf (solution): at each stage, the next edge is selected with probability proportional to the number of solutions in the subtree rooted at its lower end. By appending a correction process based on the *a posteriori* probability of the path, we saw that this procedure can be made to work even if the counter is randomised and slightly inaccurate, specifically if it is within ratio $1 + O(n^{-k_R})$, where $k_R > 0$ is a constant depending on R. However, when only rather cruder counting information is available (to within a constant factor, say) the basic "one-pass" technique breaks down owing to the accumulation of errors which are too large to be corrected.

One possible approach to coping with cruder counting estimates is to prevent the accumulation of large errors by applying a correction process at frequent intervals as we move down the tree. The effect of such a modification is to introduce an element of backtracking into the algorithm, since the correction process works by throwing away, with some appropriate probability, some final segment of the path already chosen. This suggests a more flexible dynamic approach which we now describe. The credit for proposing the following Markov chain attack on this problem goes to Mark Jerrum.

Consider again the reduction of Theorem 1.10. If the counter is exact, we may view it as assigning to each edge of the tree of derivations an integer weight equal

to the number of leaves in the subtree rooted at its lower end; the process can then be seen as a transient Markov chain in which the transition probabilities from any vertex (state) to its children are proportional to the corresponding edge weights. Suppose now that the process is no longer constrained to move downwards, but may also backtrack from any vertex to its parent, the transition probabilities to *all* adjacent vertices being proportional to the edge weights: thus we get a dynamic process in which upward and downward movements are equally likely (except at the root and the leaves). To eliminate periodicity, we also add to each state a self-loop probability of $1/2$. Viewing this process as a symmetric random walk with reflecting barriers on the *levels* of the tree, it is easy to see that it converges rapidly (essentially in time polynomial in the depth of the tree) to a stationary distribution which is uniform over levels and also uniform over leaves. Hence a short simulation of the chain generates solutions almost uniformly, and the probability of failure can be made small by repeated trials. Now suppose that we have available only an *approximate* counter for the structures in question, so that the edge weights in the tree are no longer accurate. Then we have grounds for optimism that this procedure might still work efficiently: the hope is that, since each edge weight influences transitions in both directions, the process will actually be self-correcting.

We now proceed with the details of the above construction. Let R be a self-reducible relation over the alphabet Σ, and suppose that we are given a polynomially time-bounded approximate counter C for R within ratio $\rho(n) = 1 + O(n^\beta)$ for some $\beta \in \mathbb{R}$. Thus the error ratio in C need not even be constant, but may increase polynomially with the problem size. Since R is self-reducible, C can always be modified so as to give an exact answer (which will be either 0 or 1) when its input is an atom; also, its output may always be rounded up to the nearest integer at the cost of adding at most 1 to $\rho(n)$. We shall assume throughout that C incorporates these modifications. We may also assume without loss of generality that ρ is monotonically increasing. To begin with, we shall consider the case where C is deterministic; the additional technical problems posed by randomised counters will be dealt with later.

Let $x \in \Sigma^*$ be a problem instance with $R(x) \neq \emptyset$. Our aim is to set up an ergodic Markov chain $MC(x)$ whose states are the vertices of the tree of derivations $T_R(x)$ and whose stationary distribution is uniform over the leaves of the tree. Let V, E be the vertex and edge sets respectively of $T_R(x)$, and set

$m = l_R(x)$, $r = \rho(|x|)$. Note that both m and r are polynomially bounded in $|x|$, and that the depth of the tree is at most m. For each edge $(u, v) \in E$, define the quantity

$$f(u,v) = \begin{cases} C(\text{inst}(u)), & \text{if } v \text{ is the parent of } u; \\ C(\text{inst}(v)), & \text{otherwise.} \end{cases} \tag{4.1}$$

(Recall that inst$(\cdot)$ gives the problem instance associated with any vertex in the tree.) Since C is deterministic, $f: E \to \mathbb{N}$ is a well-defined function on E. The crucial property to bear in mind is that for any edge $e \in E$, $f(e)$ approximates within ratio r the number of leaves in the maximal subtree below e.

Next we define for each vertex $v \in V$ a *degree*

$$d(v) = \sum_{u:(u,v)\in E} f(u,v)\,. \tag{4.2}$$

Note that $d(v) \geq 1$ for all $v \in V$, and that $d(v) = 1$ if v is a leaf because C is *exact* for atoms. For each ordered pair v, u of vertices, the transition probability p_{vu} from v to u is then defined to be

$$p_{vu} = \begin{cases} f(u,v)/2d(v), & \text{if } (u,v) \in E; \\ 1/2, & \text{if } u = v; \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

Thus there is a non-zero transition probability between two states iff they are adjacent in the tree. The self-loop probability $1/2$ ensures that the chain is aperiodic. It is clearly also irreducible, and hence ergodic, and it is a simple matter to verify that the stationary distribution $\pi' = (\pi_v)_{v\in V}$ is proportional to the degrees, i.e,

$$\pi_v = \frac{d(v)}{D} \qquad \forall v \in V\,,$$

where $D = \sum_{v\in V} d(v)$. Since $d(v) = 1$ for all leaves v, we immediately see that π' is uniform over them. Identifying leaves with solutions, we can therefore construct an almost uniform generator for R by applying the simulation paradigm of Figure 2.1 to the family of Markov chains $\mathcal{MC}(x)$. The generator will be fully-polynomial provided we can verify conditions (mc1)–(mc4).

The first two conditions are easy: a single call to the counter will tell us whether $R(x) = \emptyset$, and we can always start the simulation at the root of $T_R(x)$. Individual steps can be simulated by calls to the counter, which allow both the local structure of the tree and the transition probabilities (4.3) to be inferred, in

similar fashion to the algorithm of Figure 1.2. Since the size of problem instance labels in the tree never exceeds $|x|$, all calls will be polynomially bounded.

Turning our attention to condition (mc3), since non-leaf states correspond to failure of the generator we need to check that the stationary process will be found at a leaf with reasonably high probability. This is confirmed by the following lemma.

Lemma 4.1 *In the stationary distribution of $MC(x)$, the probability of being at a leaf is at least $1/2rm$.*

Proof Note that the degree sum D over the tree $T_R(x)$ may be written

$$D = \sum_{v \in V} d(v) = 2 \sum_{e \in E} f(e) .$$

Now consider the collection of edges at some fixed level of the tree. By (4.1) the weight of each such edge approximates within ratio r the number of leaves in the maximal subtree rooted at its lower end. Since these subtrees are disjoint, the aggregated weight of all edges at this level is at most $r\#R(x)$. Repeating this for all m levels yields the bound $\sum_{e \in E} f(e) \leq rm\#R(x)$, and so

$$D \leq 2rm\#R(x) . \tag{4.4}$$

Since $\pi_v = 1/D$ for each leaf v, the stationary probability of being at a leaf is

$$\sum_{\text{leaves } v} \pi_v = \frac{\#R(x)}{D} \geq \frac{1}{2rm} ,$$

as required. □

Note that m and r are each polynomially bounded in $|x|$, so the probability of being at a leaf can be boosted to $1/2$ by repeating the entire experiment only polynomially many times. This verifies (mc3).

We now address the trickier question of rate of convergence. Specifically, we want the family of chains $MC(x)$ to be rapidly mixing, as stipulated in condition (mc4). Let us see whether the characterisation of Chapter 2 helps here.

First note that the chain $\mathcal{MC}(x)$ is time-reversible by virtue of detailed balance.[1] Moreover, by (4.4) the minimum stationary state probability $\pi_{\min}^{(x)}$ satisfies

$$\pi_{\min}^{(x)} = \frac{1}{D} \geq \frac{1}{2rm\#R(x)}.$$

Since solutions are strings of length m over the fixed alphabet Σ, we have $\#R(x) \leq |\Sigma|^m$. This implies a polynomial bound of the form (2.12) on $\lg \pi_{\min}^{(x)}$. By Corollary 2.8, rapid mixing for the family $\mathcal{MC}(x)$ will therefore follow from a suitable lower bound on the conductance of the underlying graphs.

Lemma 4.2 *Let G be the underlying graph of the Markov chain $\mathcal{MC}(x)$ defined above. Then the conductance of G is bounded below by*

$$\Phi(G) \geq (4r^2m)^{-1}.$$

Proof Note first that in G each edge $(u,v) \in E$ has weight $w_{uv} = f(u,v)/2D$, while the loop at v has weight $d(v)/2D$ and all other edges have weight zero. In what follows, we will identify subsets of V with the subgraphs of $T_R(x)$ which they induce. If $S \subseteq V$ is a subtree (connected subgraph) of $T_R(x)$, we let root(S) denote the vertex of S at minimum distance from root(V), the root of $T_R(x)$, and $L(S)$ the number of leaves in S.

In order to bound the conductance of G, we claim that it suffices to consider flows out of *all* subtrees S with root$(S) \neq$ root(V). To see this, let $S \subseteq V$ be any non-empty subset of states with capacity $C_S = \sum_{v \in S} \pi_v \leq 1/2$. We may write S as the union $T_0 \cup \ldots \cup T_l$ of disjoint subtrees no pair of which are connected by an edge in $T_R(x)$, and we have

$$\Phi_S \equiv \frac{F_S}{C_S} = \frac{\sum_i F_{T_i}}{\sum_i C_{T_i}} \geq \min_i \frac{F_{T_i}}{C_{T_i}} = \min_i \Phi_{T_i}.$$

If root$(V) \notin S$, then root$(T_i) \neq$ root(V) for each i. If on the other hand root$(V) \in S$, we can work instead with $\overline{S}$ since $\Phi_S \geq \Phi_{\overline{S}}$. Thus we see that

$$\Phi(G) = \min \Phi_S,$$

---

[1] This is also an immediate corollary of the fact that $\mathcal{MC}(x)$ is a *tree process*, i.e., the edges of the underlying graph which correspond to non-zero transition probabilities form a tree.

where the minimisation is over all subtrees S of $T_R(x)$ with $\text{root}(S) \neq \text{root}(V)$.

A lower bound on Φ_S for such subtrees is readily obtained. We may assume without loss of generality that S is maximal. Then the flow out of S is just $F_S = f(\text{cut}(S))/2D$, where $\text{cut}(S)$ is the cut edge connecting S to the rest of the tree. But since $f(\text{cut}(S))$ approximates the number of leaves $L(S)$ within ratio r, the flow is bounded below by

$$F_S \geq \frac{L(S)}{2rD}. \tag{4.5}$$

On the other hand, applying the argument in the proof of Lemma 4.1 to the subtree S we may derive the bound

$$\sum_{v \in S} d(v) \leq 2 \sum_{e \in E(S)} f(e) \leq 2rmL(S), \tag{4.6}$$

where $E(S)$ is the set of edges in S. Since $C_S = \sum_{v \in S} d(v)/D$, putting (4.5) and (4.6) together yields

$$\Phi_S = \frac{F_S}{C_S} \geq \frac{1}{4r^2m},$$

which completes the proof of the lemma. □

Remark The underlying reason for the rapid convergence of $MC(x)$ should be clear from the above proof: informally, if the process is initially in some subtree S rooted below $\text{root}(V)$, then it is quite likely to emerge from S, travelling upwards, within a small number of steps. This property holds because the exit probability from S via this route is proportional to the weight of the corresponding cut edge, which cannot be too small because it is tied by the counter to the number of leaves in S. □

Lemma 4.2 ensures that the Markov chains $MC(x)$ are rapidly mixing, verifying condition (mc4). Hence the simulation paradigm yields a f.p. almost uniform generator for R. We are now in a position to state the first major result of this chapter.

Theorem 4.3 *Let R be a self-reducible relation over Σ. If there exists a polynomially time-bounded (deterministic) approximate counter for R within ratio $1 + O(n^\beta)$ for some real constant β, then there exists a f.p. almost uniform generator for R.* □

The following example is a very simple application of Theorem 4.3. We shall come to more significant applications later.

Example 4.4 Consider the relation DNFSAT defined in Chapter 1, which associates with a Boolean formula in disjunctive normal form its set of satisfying assignments. The following argument provides a crude approximate counter for DNFSAT. Let $\phi = D_1 \vee \ldots \vee D_l$ be a formula over n variables, where each D_i is a conjunction of literals, and for $1 \leq i \leq l$ let s_i be the number of assignments to the variables of ϕ which satisfy D_i. Note that the s_i are trivial to compute: if D_i contains k distinct literals and no contradictions then $s_i = 2^{n-k}$. But it is immediate that

$$\#\text{DNFSAT}(\phi) \leq \sum_{i=1}^{l} s_i \leq l\#\text{DNFSAT}(\phi)\,,$$

so that $\sum s_i$ approximates the number of satisfying assignments of ϕ within the polynomial ratio l. By Theorem 4.3, this gives us a f.p. almost uniform generator, and by Theorem 1.14 a f.p. randomised approximate counter for DNFSAT.

As mentioned in Chapter 1, such algorithms have been constructed directly by Jerrum, Valiant and Vazirani [33] and Karp and Luby [34] respectively. □

If the approximate counter in Theorem 4.3 is *randomised*, so that it may occasionally produce arbitrarily bad results, a similar reduction still goes through but at the cost of some tiresome technicalities. We summarise the proof in this case.

Theorem 4.5 *The result of Theorem 4.3 still holds even if the approximate counter for R is randomised.*

Proof (sketch) Let x be a problem instance for which $R(x) \neq \emptyset$. As before, assume ρ is monotonic and set $m = l_R(x)$, $r = \rho(|x|)$. We begin by considering the intermediate case where the counter C is randomised but always produces estimates which are within ratio r of their correct values. We again define a Markov chain $MC(x)$ on the tree $T_R(x)$, whose transition probabilities are now determined as follows. Suppose the process is currently at vertex v, and let U be the set of children of v. For each $u \in U \cup \{v\}$, make a call $C(\text{inst}(u))$ to the counter and denote the result $c(u)$; then make a further independent set of

calls $C(\mathrm{inst}(u))$ for the same vertices u and denote their sum $d(v)$. Finally, make a transition to an adjacent vertex u with probability

$$\begin{cases} c(u)/4r^2 d(v), & \text{if } u \text{ is a child of } v; \\ c(v)/4r^2 d(v), & \text{if } u \text{ is the parent of } v, \end{cases} \tag{4.7}$$

and remain at v otherwise. (Note that the factor $1/4r^2$ ensures that these transitions are always well-defined, and that there is a self-loop probability of at least $1/2$ in each state; we have used $1/4$ rather than $1/2$ for consistency with the second part of the proof.) Clearly, if C is deterministic this reduces (except for a uniform factor of $1/2r^2$) to the original chain. In the randomised case, it is easy to see that the transition probability p_{vu} from v to u is actually the expectation

$$\mathrm{E}\left(\frac{f(u,v)}{4r^2 d(v)}\right) = \frac{1}{4r^2}\,\mathrm{E}\big(f(u,v)\big)\,\mathrm{E}\left(\frac{1}{d(v)}\right),$$

where the random variable $f(u,v)$ is defined as in (4.1) and is independent of $d(v)$. The stationary distribution π' therefore satisfies

$$\pi_v \propto 1/\mathrm{E}\big(d(v)^{-1}\big) \qquad \forall v \in V\,,$$

and the fact that C is exact for atoms implies that $d(v) = 1$ with probability 1 for leaves v. The chain is clearly still time-reversible, and the rest of the proof goes through essentially as in the deterministic case, with $d(v)$ and $f(u,v)$ replaced by $1/\mathrm{E}\big(d(v)^{-1}\big)$ and $\mathrm{E}\big(f(u,v)\big)$ respectively.

Now suppose that the counter may in addition produce arbitrarily bad results with some small probability δ: by Lemma 1.2 we may assume that $\delta \leq 2^{-p(|x|)}$ for all problem instances in the tree, where p is any desired polynomial, while C still runs in time polynomial in $|x|$. Since we are no longer able to infer the structure of $T_R(x)$ with certainty, we must now work in the larger self-reducibility tree $\tilde{T}_R(x)$ (c.f. Section 1.1). We let V, $\tilde{V}$ denote the vertex sets of $T_R(x)$ and $\tilde{T}_R(x)$ respectively. Note that $\tilde{V} \setminus V$ consists of a union of disjoint maximal subtrees of $\tilde{T}_R(x)$. Some modifications to the transition probabilities are also necessary. At vertex $v \in \tilde{V}$, we compute values $c(u)$, $u \in U \cup \{v\}$, and $d(v)$ as before, where now U is the set of children of v in $\tilde{T}_R(x)$. If $d(v) = 0$ then we make a transition to the parent of v (if it exists) with probability $1/4$, and remain at v with probability $3/4$. Otherwise, we test whether $\sum_u c(u) > 4r^2 d(v)$: if so, we remain at v; if not, we make a transition to a neighbouring vertex with probabilities as in (4.7). Once again, the leaves of $T_R(x)$ are treated as a special case.

This chain is clearly ergodic on some subset of $\tilde{V}$ containing V, namely those states which communicate with the root. Henceforth we redefine $\tilde{V}$ to include only such states. The chain is also still time-reversible because it is a tree process. Let us first observe that the new vertices in $\tilde{V} \setminus V$ have negligible effect. All transitions from V to $\tilde{V} \setminus V$ occur with at most tiny probability δ, so if started in V the process is unlikely to leave V during the course of the simulation. Should it enter a subtree in $\tilde{V} \setminus V$, however, the random variable $d(v)$ at the root vertex v will take the value 0 with probability very close to 1, thus causing the chain to leave the subtree rapidly. In fact, it is not hard to see that the stationary probability π_v of a vertex $v \in \tilde{V} \setminus V$ is at most $O(\delta^k)$, where k is the distance of v from V in $\tilde{T}_R(x)$. As a result, the total weight of $\tilde{V} \setminus V$ in the stationary distribution is small. Furthermore, the large exit probability from subtrees $S \subseteq \tilde{V} \setminus V$ ensures a lower bound on Φ_S similar to that in the proof of Lemma 4.2.

Examination of the transition probabilities *within* V reveals that we can view this portion of the chain as a chain of the restricted kind described in the first part of the proof whose transition probabilities have been perturbed by a factor in the range $(1 \pm \delta')$, where δ' depends on δ and can be made exponentially small in $|x|$. It is then easy to see that the stationary probabilities of states in V undergo similarly small perturbations in the range $(1 \pm \delta')^m$. As a result, a lower bound as in the proof of Lemma 4.2 also holds for subtrees S with $\mathrm{root}(S) \in V$, and so for all subtrees, which again implies that the conductance $\Phi(G)$ is suitably bounded below. Assuming that the simulation starts at the root, we therefore get rapid convergence over the subset V of the state space,[2] which is sufficient since V includes all leaves of $T_R(x)$. A test applied to leaf labels ensures that no non-solutions are output. □

Remark There is actually a much simpler way to prove Theorem 4.5, though the resulting algorithm is rather less natural and the process is no longer strictly a Markov chain. Note that the simulation of Theorem 4.3 can still be performed

[2]More precisely, we are using the r.p.d. $\Delta_V(t)$ over V here, as defined in Chapter 2. Note that Theorem 2.5 implies a sufficient condition for rapid mixing with respect to this measure also.

using a randomised counter if we arrange to *remember* the outputs of the counter on all previous calls so that each edge weight is computed at most once. Provided all values returned by the counter are accurate within the given ratio, we are effectively in the situation of Theorem 4.3 and our earlier analysis applies. By powering the counter, we can ensure that this condition fails to hold with vanishingly small probability, so the effect on the overall process will be negligible. □

A direct consequence of Theorem 4.5 is a powerful bootstrapping theorem for approximate counters. Recall from Theorem 1.14 that a f.p. almost uniform generator for a self-reducible relation R can be used to construct a f.p. randomised approximate counter for R. Thus, starting from an approximate counter within ratio $1 + O(n^\beta)$, and proceeding via the reduction described in this section, we arrive at a counter with arbitrarily good asymptotic behaviour.

Theorem 4.6 *Let R be a self-reducible relation over Σ. If there exists a polynomially time-bounded randomised approximate counter for R within ratio $1+O(n^\beta)$ for some real constant β, then there exists a f.p. randomised approximate counter for R.* □

The chief significance of Theorem 4.6 is that it establishes a notion of approximate counting which is *robust* with respect to polynomial time computation, at least for the large class of self-reducible relations. This is evident from the following:

Corollary 4.7 *Let $R \subseteq \Sigma^* \times \Sigma^*$ be self-reducible. Then there exists a polynomially time-bounded randomised approximate counter for R within ratio $1 + O(n^\beta)$ either for all $\beta \in \mathbb{R}$ or for no $\beta \in \mathbb{R}$.* □

We are therefore justified in calling the counting problem for a self-reducible relation R *tractable* iff there is a polynomial time randomised procedure which with high probability estimates $\#R(x)$ to within a factor of the form $1+O(|x|^\beta)$, for some fixed real β. This provides a particularly convenient means of classifying #P-complete counting problems according to a simple criterion of approximability. We conjecture that this notion will prove useful in the future classification of counting problems.

It is natural to ask whether a similar state of affairs holds in the case of generation problems. Unfortunately, we know of no analogue of Theorem 4.6 for improving almost uniform generators with very large (polynomial) bias. Nevertheless, Theorem 4.5 can be used to obtain a weaker bootstrapping theorem for generators.

Theorem 4.8 *Let $R \subseteq \Sigma \times \Sigma^*$ be self-reducible and k_R be a constant as in Section 1.1 such that, for all pairs $\langle x, y \rangle \in R$, $|y| = O(|x|^{k_R})$. If there exists a polynomially time-bounded almost uniform generator for R within tolerance $O(n^{-k_R})$ then there exists a f.p. almost uniform generator for R.*

Proof We know from Theorem 1.16 that a generator satisfying the above hypothesis can be used to construct a polynomially time-bounded randomised approximate counter for R within ratio $O(n^\beta)$ for some constant β. An application of Theorem 4.5 now yields the result. □

On reflection, Theorem 4.8 is quite remarkable: it says that, by observing only polynomially many outputs of a black box generator whose bias is a fixed function of the input size, we are effectively able to reduce this bias. Indeed, the reduction is huge, from $O(n^{-k_R})$ to the exponential factor $\exp(-n^\beta)$ for any desired real β. This seems counter-intuitive since such an experiment can apparently yield only very local information about the output distribution. The non-trivial reduction of Theorem 4.5 seemingly plays a crucial rôle here.

We close this section with an updated version of Corollary 1.15 which summarises what we know about approximate counting and almost uniform generation for self-reducible relations.

Corollary 4.9 *For a self-reducible relation R over Σ, the following are equivalent:*

(i) *There exists a f.p. almost uniform generator for R.*

(ii) *There exists a f.p. randomised approximate counter for R.*

(iii) *There exists a polynomially time-bounded randomised approximate counter for R within ratio $O(n^\beta)$ for some real constant β.*

(iv) *There exists a polynomially time-bounded almost uniform generator for R within tolerance $O(n^{-k_R})$, where k_R is a constant as above.* □

4.2 Self-embeddable relations

This short section is in the nature of a caveat to the bootstrapping results we have just derived. It turns out that similar results hold for a rather trivial reason if the relation R in question has a property which we might term *self-embeddability*. Informally, R is self-embeddable if there exists a polynomial time computable function ξ which takes a pair x_1, x_2 of problem instances and "embeds" them in an instance $\xi(x_1, x_2)$, whose size is at most linear in $|x_1|$ and $|x_2|$ and whose solution set is in (1-1)-correspondence with the product set $R(x_1) \times R(x_2)$. We also demand that, from each solution for $\xi(x_1, x_2)$, the corresponding pair of solutions for x_1 and x_2 can be recovered easily. The relation MATCH of Example 1.1 is clearly self-embeddable: for any pair G_1, G_2 of graphs we may take $\xi(G_1, G_2)$ to be the disjoint union of G_1 and G_2. A slightly less trivial example is the following:

Example 4.10 Consider the relation which associates with a directed graph G its set of (directed) Hamiltonian paths. Then given a pair G_1, G_2 of graphs, the required embedding function ξ adds to their disjoint union a new vertex v, together with edges from v to all vertices of G_1 and from all vertices of G_2 to v. □

For self-embeddable relations, there is a simple bootstrapping mechanism for approximate counters. (This observation is due to Keith Edwards. It is also implicit in the work of several other authors, such as Stockmeyer [54].)

Theorem 4.11 *Let $R \subseteq \Sigma^* \times \Sigma^*$ be self-embeddable. If there exists a polynomially time-bounded (randomised) approximate counter for R within ratio $1 + O(n^\beta)$ for some real constant β, then there exists a f.p. (randomised) approximate counter for R.*

Proof Let C be a counter for R as above within ratio $1+\rho'(n)$, where $\rho'(n) = cn^\beta$ and $c, \beta > 0$. Given a problem instance $x \in \Sigma^*$ and a positive real $\epsilon \leq 1$, apply

the embedding construction $q = \lceil \lg(2cn^\beta/\epsilon) \rceil$ times to obtain an instance z with $\#R(z) = \#R(x)^{2^q}$. From the form of q, $|z|$ is bounded by a polynomial in $|x|$ and ϵ^{-1}. Now use C to approximate $\#R(z)$, and take the 2^qth root of the result. The approximation ratio for the final estimate is then at most $(1 + cn^\beta)^{1/2^q} \leq 1 + \epsilon$, as required. □

Theorem 4.11 does not undermine the contribution of the previous section, for several reasons. Firstly, although many natural relations are self-embeddable, there seem to be a number of significant exceptions among self-reducible relations, including DNF-satisfiability and important *restricted* versions of familiar relations, such as Hamiltonian paths in planar graphs. Moreover, the Markov chain reduction technique presented earlier is quite natural and general and can sometimes be applied even in the absence of self-reducibility. Evidence for this is provided by the relation GRAPHS discussed in the next section, which is apparently neither self-embeddable nor self-reducible under the degree restrictions imposed there.

How does self-embeddability impact upon almost uniform generation? Currently, the best answer we can give to this question is that, in the presence of both self-reducibility and self-embeddability, a very strong bootstrapping mechanism for almost uniform generators is available: this is the subject of our next theorem. However, we know of no useful result when the self-reducibility assumption is dropped.

Theorem 4.12 *Let $R \subseteq \Sigma^* \times \Sigma^*$ be both self-reducible and self-embeddable. If there exists a polynomially time-bounded almost uniform generator for R within tolerance $O(n^\beta)$ for some real constant β, then there exists a f.p. almost uniform generator for R.*

Proof Suppose the given generator has tolerance at most $cn^\beta - 1$ for $c, \beta > 0$. We show how to construct a f.p. randomised approximate counter for R: the result will then follow from Theorem 1.10.

The construction is essentially that of Theorem 1.14 (see Figure 1.3 and the accompanying discussion—we shall adopt the notation used there in what follows). This shows how to estimate the number of leaves in the tree of derivations given

a means of sampling them with small bias. We need therefore only show how to use our very biased generator to produce such a sample.

Given a problem instance x, the trick is again to exploit the self-embeddability of R in creating a new problem instance z with $R(z) \approx R(x)^t$ for some suitably large t. We may then regard solutions to z as t-samples of solutions to x. The key fact is that, for large enough t, almost all such samples are good enough for the procedure of Figure 1.3, so that even the very large bias in generating a sample has negligible effect.

To make this precise, let us call a t-sample *good* for the algorithm of Figure 1.3 if it yields an estimate of the proportion of leaves in the corresponding subtree within ratio $1+\epsilon/2m$. The correct operation of the algorithm depends on samples being good with probability at least $1-1/8m$. Now by Proposition 1.12, setting $\xi = \epsilon/4m$ and $\delta = (8c^2n^{2\beta}m)^{-1}$, we see that if $t \geq 9d^3/\xi^2\delta$ then a *uniformly* selected t-sample will be good with probability at least $1-\delta$. However, we are only able to select samples almost uniformly within tolerance cn^β, so we have

$$\Pr\big(\text{sample is not good}\big) \leq c^2n^{2\beta}\delta = 1/8m,$$

as required. The bound on t means that a sample size which is polynomially bounded in $|x|$ and ϵ^{-1} will do. Generator failure is handled as before using Proposition 1.13. □

4.3 Graphs with specified degrees

So far in this chapter we have been concerned with the general implications of the reduction from generation to counting of Section 4.1. In this section, we illustrate its practical value by applying it to a specific problem which is of independent interest. The application will also indicate how the requirement of self-reducibility may be relaxed in practice.

Specifically, we shall be concerned with the following question: given a sequence $\mathbf{g} = (g_0, \ldots, g_{n-1})$ of non-negative integers, is it possible to efficiently generate labelled graphs with vertex set $\{0, 1, \ldots, n-1\}$ in which vertex i has degree g_i, $0 \leq i \leq n-1$, such that each graph occurs with roughly equal probability? We also allow a set X of excluded edges to be specified, i.e., the graphs

must all be subgraphs of some given graph. Using the ideas of Section 4.1, we are able to answer this question affirmatively provided that the maximum degree in the problem does not grow too rapidly with the number of vertices n.

The special case of this problem in which X is empty and the graphs are regular, i.e., $g_i = k$ for all i and some k, is of particular interest and has been considered by several authors. Regular graphs are a natural class to study in their own right, and have recently become an important model in the theory of random graphs (see [11] for a definitive treatment of this field). A generation procedure for the above problem would provide a means of examining "typical" regular graphs with a given number of vertices and given degree and investigating their properties, about many of which little is known. Furthermore, it has recently been shown by Wormald [63] that generation techniques for labelled graphs with a given degree sequence can be used in the uniform generation of *unlabelled* regular graphs.

Wormald [62] gives an efficient algorithm for uniformly generating labelled cubic graphs on n vertices. However, this is based on specific recurrence relations for the associated counting problem and does not generalise easily to higher degrees. A simpler method proposed in [62], and also by Bollobás and Thomason [11], uniformly generates labelled regular graphs of arbitrary degree k, but the probability of failure remains polynomially bounded only if $k = \mathrm{O}\left((\log n)^{1/2}\right)$. When the degree is permitted to increase more rapidly with n, it seems difficult to generate the graphs with anything approaching equal probabilities. (In the work of Tinhofer [55], for example, the probabilities associated with different graphs may vary widely.) Our method, which appears to rely on the full power of the reduction to counting developed in Section 4.1, requires only that $k = \mathrm{O}(n^{1/3})$ and achieves a distribution over the graphs which is asymptotically very close to uniform.

In keeping with our general approach, we begin by defining a relation which describes the graphs of interest. A (*labelled*) *degree sequence* on vertex set $[n] = \{0, \ldots, n-1\}$ is a sequence $\mathbf{g} = (g_0, \ldots, g_{n-1})$ of non-negative integers such that $\sum_i g_i = 2e(\mathbf{g})$ is even, and a *graph on* $\mathbf{g}$ is a graph with vertex set $[n]$ in which vertex i has degree g_i, $0 \leq i \leq n-1$. (All graphs here are assumed to be simple and undirected.) If the vertex set is understood, we shall identify a graph with its edge set. As mentioned earlier, we also allow a set of forbidden edges to be specified. Accordingly, we define the relation GRAPHS which associates with

each problem instance of the form $\langle \mathbf{g}, X \rangle$, where $\mathbf{g}$ is a degree sequence on $[n]$ and X is a labelled graph with vertex set $[n]$, the solution set

$$\text{GRAPHS}(\mathbf{g}, X) = \{ G : G \text{ is a graph on } \mathbf{g} \text{ having no edge in common with } X \}.$$

We refer to X as an *excluded graph* for $\mathbf{g}$. Although this relation is self-reducible as it stands, we get a more symmetrical structure using the relation R defined by

$$R(\mathbf{g}, X) = \{ \langle G, \omega \rangle : G \in \text{GRAPHS}(\mathbf{g}, X) \text{ and } \omega \text{ is an edge-ordering of } G \}.$$

Clearly, we can move freely between these relations since any solution set $R(\mathbf{g}, X)$ contains precisely $e(\mathbf{g})!$ ordered copies of each element of $\text{GRAPHS}(\mathbf{g}, X)$.

Next we specify a self-reducibility on R by defining the tree of derivations $T_R(\mathbf{g}, X)$, assuming that $R(\mathbf{g}, X) \neq \emptyset$. In this tree, the object $\langle G, \omega \rangle$ will be derived by successively adding the edges of G in the order determined by ω. More precisely, the partial solution labels of the tree are in (1-1)-correspondence with pairs $\langle \overline{H}, \omega \rangle$ in which $\overline{H}$ is a graph with vertex set $[n]$ which can be extended to at least one graph in $\text{GRAPHS}(\mathbf{g}, X)$, and ω is an edge-ordering of $\overline{H}$. The root has label $\langle \emptyset, \emptyset \rangle$, while the children of the vertex with label $\langle \overline{H}, \omega \rangle$ have labels of the form $\langle \overline{H} \cup \{(i,j)\}, \omega + (i,j) \rangle$ for some edge (i,j), where $\omega + (i,j)$ denotes the extension of ω in which (i,j) is the largest element. The problem instance label of a vertex v is determined by its partial solution label $\langle \overline{H}, \omega \rangle$ as follows. Let $\overline{\mathbf{h}} = (\overline{h}_0, \ldots, \overline{h}_{n-1})$ be the degree sequence of $\overline{H}$, and define $\mathbf{h} = \mathbf{g} - \overline{\mathbf{h}}$, where the subtraction is pointwise. Also, let Y be the subgraph of $X \cup \overline{H}$ obtained by deleting all edges (i,j) for which either $\overline{h}_i = g_i$ or $\overline{h}_j = g_j$. Then the problem instance label of v is $\langle \mathbf{h}, Y \rangle$.

Note that the deletion of redundant constraints from $X \cup \overline{H}$ is not necessary for the consistency of the tree, but it will prove useful later—in the proof of Lemma 4.16—that Y represents only the *essential* excluded graph. From now on, we will in fact assume without loss of generality that all problem instances $\langle \mathbf{g}, X \rangle$ have had redundant constraints removed. In particular, this means that the problem instance label of the root of the tree is just $\langle \mathbf{g}, X \rangle$. It also justifies our use of $e(\mathbf{g})$ as a measure of input size for this problem when stating approximation results in the sequel.

Now that we have a tree of derivations for R, the reduction of Section 4.1 will give us a fully-polynomial almost uniform generator for R (and hence for

GRAPHS) provided that we can count these structures with sufficient accuracy. The counting problem for GRAPHS has received much attention over a number of years, where the aim has chiefly been to extend the validity of asymptotic estimates to a wider range of degrees (see [40] for a brief survey). The best result available to date is due to McKay, and we quote this below.

Given a degree sequence g on $[n]$ and an excluded graph X for g, let $\mathbf{x} = (x_0, \ldots, x_{n-1})$ be the degree sequence of X, and define $\gamma(\mathrm{g}, X) = \max\{g_{\max}^2, g_{\max} x_{\max}\}$, where $g_{\max} = \max_i g_i$ and $x_{\max} = \max_i x_i$. We shall use γ to express bounds on the degrees involved in the problem. Furthermore, if $g_{\max} > 0$, set

$$\lambda(\mathrm{g}) = \frac{1}{4e(\mathrm{g})} \sum_{i=0}^{n-1} g_i(g_i - 1)\,; \quad \mu(\mathrm{g}, X) = \frac{1}{2e(\mathrm{g})} \sum_{(i,j)\in X} g_i g_j\,.$$

Theorem 4.13 (McKay [40]) *There exists a positive constant r_0 with the property that, for any problem instance $\langle \mathrm{g}, X\rangle$ with $g_{\max} > 0$, and $\gamma(\mathrm{g}, X) \le e(\mathrm{g})/10$, the quantity*

$$\frac{(2e(\mathrm{g}))!}{e(\mathrm{g})!\, 2^{e(\mathrm{g})} \prod_{i=0}^{n-1} g_i!} \exp\left(-\lambda(\mathrm{g}) - \lambda(\mathrm{g})^2 - \mu(\mathrm{g}, X)\right) \tag{4.8}$$

approximates #GRAPHS(g, X) *within ratio* $\exp\left(r_0 \gamma(\mathrm{g}, X)^2 / e(\mathrm{g})\right)$. □

Remarks (a) Actually, McKay's result is slightly stronger than this: we have stated it in a simplified form which is adequate for our purposes.

(b) The estimate in Theorem 4.13 immediately leads to a simple method, suggested by Bollobás and Thomason [11] and Wormald [62], for generating graphs whose degrees grow slowly with the number of edges: make g_i copies of vertex i for each i, generate a *pairing* (i.e., a perfect matching in the complete graph on these vertices) uniformly at random, and then collapse the copies to a single vertex again. The result will be a multigraph on g, and the distribution over GRAPHS(g, X) is uniform, but the procedure may fail since not all the graphs generated in this way will be simple or avoid X. The exponential factor in (4.8) can be interpreted as (an approximation of) the probability that a randomly chosen pairing yields an element of GRAPHS(g, X). It is then clear from the definitions of λ and μ that, provided $\gamma(\mathrm{g}, X) = \mathrm{O}(\log e(\mathrm{g}))$, this probability is polynomially bounded below, so that the method is effective in this range. For regular graphs, this implies a degree bound of $\mathrm{O}\left((\log n)^{1/2}\right)$. □

Let us now restate Theorem 4.13 in a more convenient form.

Corollary 4.14 *Let Q, B be real numbers with $Q > 0$ and $B \geq 100Q^4$. Then for all problem instances $\langle \mathrm{g}, X \rangle$ for which either $e(\mathrm{g}) \leq B$ or $\gamma(\mathrm{g}, X) \leq Q^2 e(\mathrm{g})^{1/2}$, the quantity $\#R(\mathrm{g}, X)$ can be approximated in polynomial time within a constant ratio.*

Proof We have already observed that $\#R(\mathrm{g}, X) = e(\mathrm{g})! \#\textsc{graphs}(\mathrm{g}, X)$, so we need only estimate the latter. Note that when $e(\mathrm{g}) > B$ the bound on γ ensures also that $\gamma(\mathrm{g}, X) \leq e(\mathrm{g})/10$, so we may appeal to Theorem 4.13. The expression in (4.8) can clearly be evaluated in polynomial time, and yields an approximation within the constant ratio $\exp(r_0 Q^4)$ in all relevant cases, except when $g_{\max} = 0$ or possibly when $e(\mathrm{g}) \leq B$. The first case is trivial; to handle the second, note that for fixed B there are only a constant number of instances, up to relabelling of the vertices, for which $e(\mathrm{g}) \leq B$, so all counting in this range may be done exactly by explicit enumeration. (Alternatively, in practice any convenient approximation method may be used, subject to the proviso that it yields the answer 0 iff $\#\textsc{graphs}(\mathrm{g}, X) = 0$: this property can be tested in polynomial time using matching techniques.) □

Now let us see whether Corollary 4.14 is powerful enough to allow us to construct a generation algorithm for GRAPHS via the reduction to counting embodied in Theorem 4.3. Ideally, we might hope to handle instances for which $\gamma(\mathrm{g}, X)$ grows as $\mathrm{O}\left(e(\mathrm{g})^{1/2}\right)$. However, this does not follow immediately since the relation R is no longer self-reducible when restricted in this way. In other words, even if $g_{\max}$ and $x_{\max}$ are suitably bounded, the tree $T_R(\mathrm{g}, X)$ will in general contain vertices whose problem instances $\langle \mathrm{h}, Y \rangle$ are *unbalanced* in the sense that the degrees are rather large compared to the number of edges $e(\mathrm{h})$, so that we cannot guarantee reasonable counting estimates over the whole tree. We will overcome this problem by naïvely *pruning* the tree in such a way as to leave only problem instances which do fall within the bounds of Corollary 4.14, though we will have to do a little work to check that the effects of this are not too drastic.

For any pair Q, B of real numbers with $Q > 0$ and $B \geq 100Q^4$, we call a problem instance $\langle \mathrm{g}, X \rangle$ (Q, B)-*balanced* if either $e(\mathrm{g}) \leq B$ or $\gamma(\mathrm{g}, X) \leq Q^2 e(\mathrm{g})^{1/2}$.

If $\langle \mathbf{g}, X \rangle$ is (Q, B)-balanced and $R(\mathbf{g}, X) \neq \emptyset$, then the *pruned* tree $T_R^{(Q,B)}(\mathbf{g}, X)$ with respect to Q, B is obtained by deleting from $T_R(\mathbf{g}, X)$ each vertex whose problem instance label is not (Q, B)-balanced, together with the entire subtree rooted at the vertex.

Suppose now that we have fixed a pair Q, B as above. Then for any (Q, B)-balanced instance $\langle \mathbf{g}, X \rangle$ with $\text{GRAPHS}(\mathbf{g}, X) \neq \emptyset$, we may define a time-reversible Markov chain $MC(\mathbf{g}, X)$ on the pruned tree in precisely the same manner as in Section 4.1, using the counting estimates of Corollary 4.14. Since all remaining problem instance labels are (Q, B)-balanced, the ratio in the approximation will be bounded by a constant in all cases. Once again, each vertex v will have an associated "degree" $d(v)$ as in (4.2), and the stationary distribution will be proportional to the degrees. Moreover, since we have deleted some subtrees and otherwise left the transition probabilities unaltered, it should be clear that the conductance is still bounded as in Lemma 4.2, which in turn implies that the rapid mixing property holds. (This merely reflects the intuitive fact that the removal of extremal portions of a Markov chain cannot slow down the rate of convergence on the remainder of the state space.) We may therefore state the following fact:

Lemma 4.15 *For any fixed Q, B as above, and all (Q, B)-balanced problem instances $\langle \mathbf{g}, X \rangle$ with $\text{GRAPHS}(\mathbf{g}, X) \neq \emptyset$, the family of Markov chains $MC(\mathbf{g}, X)$ is rapidly mixing.* □

Just as before, the stationary distribution can be made uniform over the leaves by ensuring exact counting estimates at this level. However, since we have lost some leaves by pruning, it is by no means obvious that the *induced* distribution on $\text{GRAPHS}(\mathbf{g}, X)$ obtained by forgetting the edge orderings is even close to uniform, or that the failure probability is still bounded. Both these facts will follow from the lemma below, which says that in the pruning process we lose at most a small fraction of the leaves corresponding to any graph in $\text{GRAPHS}(\mathbf{g}, X)$, provided that the constants Q, B are suitably chosen.

Lemma 4.16 *Let $\mathcal{F}$ be a family of problem instances $\langle \mathbf{g}, X \rangle$ satisfying the bound $\max\{g_{\max}, x_{\max}\} = O\left(e(\mathbf{g})^{1/4}\right)$, and β a real constant. Then there exists a pair of real numbers Q, B as above (which depend on $\mathcal{F}$ and β) such that, for each instance $\langle \mathbf{g}, X \rangle \in \mathcal{F}$ with $\text{GRAPHS}(\mathbf{g}, X) \neq \emptyset$, and each $G \in \text{GRAPHS}(\mathbf{g}, X)$,*

the pruned tree $T_R^{(Q,B)}(\mathrm{g}, X)$ *contains at least* $e(\mathrm{g})!\left(1 - e(\mathrm{g})^{-\beta}/4\right)$ *leaves with solution label* G.

We postpone the rather technical proof of this lemma until we have examined its consequences, which constitute the central results of this section.

Theorem 4.17 *For any fixed real* β, *there exists a polynomially time-bounded almost uniform generator for* GRAPHS *within tolerance* $e(\mathrm{g})^{-\beta}$, *provided that the degrees involved are bounded as* $\max\{g_{\max}, x_{\max}\} = \mathrm{O}\left(e(\mathrm{g})^{1/4}\right)$.

Proof We assume without loss of generality that $\beta \geq 0$ and that $e(\mathrm{g}) > 0$. Let Q, B be real numbers satisfying the conditions of Lemma 4.16 for the given value of β. Assuming that $\mathrm{GRAPHS}(\mathrm{g}, X) \neq \emptyset$, simulate the Markov chain $MC(\mathrm{g}, X)$ as defined above. By Lemma 4.15, a polynomially bounded simulation is sufficient to ensure a r.p.d. of at most $e(\mathrm{g})^{-\beta}/6$. But by Lemma 4.16, the stationary distribution of the chain induces a distribution over $\mathrm{GRAPHS}(\mathrm{g}, X)$ which is almost uniform within tolerance $e(\mathrm{g})^{-\beta}/2$, since $e(\mathrm{g}) > 0$ and $\beta \geq 0$. The overall tolerance is then at most $e(\mathrm{g})^{-\beta}$, as required. Finally, again by Lemma 4.16, the stationary probability of being at a leaf is bounded below as in Lemma 4.1 except for an additional factor due to pruning of $\left(1 - e(\mathrm{g})^{-\beta}/4\right) \geq 3/4$. □

Corollary 4.18 *For any fixed real* β, *there exists a polynomially time-bounded almost uniform generator for labelled* k*-regular graphs on* n *vertices within tolerance* $n^{-\beta}$, *provided that the degree is bounded as* $k = \mathrm{O}(n^{1/3})$. □

We could of course allow the tolerance in the above algorithm to be specified as part of the input. However, there is no reason to suppose that the resulting generator would be fully-polynomial since we can say nothing useful about the behaviour of the counter in Corollary 4.14 for "small" instances as Q and B vary. Thus the polynomial bias claimed here is apparently the best we can achieve in polynomial time. Note that the source of the bias is essentially just the pruning operation on the tree: the effect of the truncation of the Markov chain is exponentially small as in Theorem 4.3, and thus negligible by comparison.

It remains now for us to prove Lemma 4.16. For this we require a preliminary technical result.

Proposition 4.19 *Let Z be a random variable denoting the number of green objects in a random sample (without replacement) of size $s > 0$ from a population of size $m \geq 2s$ made up of g green and $b = m - g$ blue objects, and let $\mu = \mathrm{E}(Z) = sg/m$. Then for any real $\alpha > 0$,*

$$\Pr(Z > \alpha\mu) \leq s\left(\frac{2\mathrm{e}}{\alpha}\right)^{\alpha\mu}.$$

Proof Note that Z is distributed hypergeometrically with mean $\mathrm{E}(Z) = \mu$ as claimed. Now set $r = \alpha\mu$. If $r < sg/(m-s)$ then the right-hand side of the above inequality is greater than 1 and there is nothing to prove. Assume therefore that $r \geq sg/(m-s)$. For each i, $1 \leq i \leq s$, the probability that the ith choice yields a green object, conditional on the preceding choices, certainly cannot exceed $g/(m-s)$, since there are always at least $m-s$ elements remaining in the pool. Thus for any $r' \in \mathbb{N}$ with $r' \geq r$ we have

$$\Pr(Z = r') \leq q_{r'} = \binom{s}{r'}\left(\frac{g}{m-s}\right)^{r'}.$$

But we have also

$$\binom{s}{r'} = \frac{s!}{r'!(s-r')!} \leq \frac{s^{r'}}{r'!} \leq \left(\frac{\mathrm{e}s}{r'}\right)^{r'},$$

by Stirling's approximation, so that

$$q_{r'} \leq \left(\frac{\mathrm{e}sg}{r'(m-s)}\right)^{r'}.$$

Now the function $f(x) = (c/x)^x$, $(c \in \mathbb{R}^+)$, is monotonically decreasing for $c/x \leq \mathrm{e}$; hence, since $r' \geq r \geq sg/(m-s)$, we have the bound

$$q_{r'} \leq \left(\frac{\mathrm{e}sg}{r(m-s)}\right)^{r} \leq \left(\frac{2\mathrm{e}}{\alpha}\right)^{\alpha\mu} \qquad \forall r' \geq r,$$

and consequently

$$\Pr(Z > r) \leq \sum_{r'=\lceil r \rceil}^{s} \Pr(Z = r') \leq s\left(\frac{2\mathrm{e}}{\alpha}\right)^{\alpha\mu},$$

as required. □

Proof of Lemma 4.16 By virtue of the asymptotic bounds on $g_{\max}$ and $x_{\max}$, we may choose $Q > 0$ such that $\max\{g_{\max}, x_{\max}\} \leq (Q/4)e(\mathbf{g})^{1/4}$ for all instances in the family. This implies a lower bound on B of $100Q^4$: further constraints on B will be introduced below. Note that all instances in the family are certainly (Q, B)-balanced.

For problem instances with $e(\mathbf{g}) \leq B$, there is nothing to prove as no pruning takes place in the tree $T_R(\mathbf{g}, X)$. So let $\langle \mathbf{g}, X \rangle$ be an instance in the family with $m = e(\mathbf{g}) > B$, and G be any graph in GRAPHS$(\mathbf{g}, X)$, assumed non-empty. In order to estimate the proportion of all $m!$ derivations of G present in the pruned tree $T_R^{(Q,B)}(\mathbf{g}, X)$, we estimate the probability that a *randomly chosen* derivation of G is present. More precisely, consider the random process $\left(\overline{H}^{(t)}\right)_{t=0}^{m}$, where $\overline{H}^{(0)} = \varnothing$ and, for $t \geq 1$, $\overline{H}^{(t)}$ is a subgraph of G having precisely t edges which is obtained from $\overline{H}^{(t-1)}$ by adding a single edge, all unused edges of G being equiprobable. If we identify $\overline{H}^{(t)}$ with a problem instance label $\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle$ in the tree of derivations as before, then a random derivation $\left(\overline{H}^{(t)}\right)$ is still present after pruning iff $\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle$ is (Q, B)-balanced for $0 \leq t \leq m$. (Recall that $\langle \mathbf{h}^{(0)}, Y^{(0)} \rangle = \langle \mathbf{g}, X \rangle$ is always (Q, B)-balanced by choice of Q.) The proportion of all $m!$ derivations of G which are present after pruning is therefore just

$$\Pr\left(\bigwedge_{t=0}^{m} \left(\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle \text{ is } (Q, B)\text{-balanced}\right)\right). \tag{4.9}$$

We proceed to obtain a lower bound on (4.9) by showing that, for each t separately, $\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle$ is almost surely (Q, B)-balanced, provided we make B large enough. Note that this is just the event that the problem instance corresponding to a randomly chosen t-edge subgraph of G is (Q, B)-balanced. The proof divides into four stages, corresponding to various ranges of values of t.

(i) *If* $0 \leq t \leq m/2$, *then* $\Pr\left(\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle \text{ is } (Q, B)\text{-balanced}\right) = 1$.
For any t in this range, we must have $h_{\max}^{(t)} \leq g_{\max}$ and $y_{\max}^{(t)} \leq x_{\max} + g_{\max}$, so that $\gamma(\mathbf{h}^{(t)}, Y^{(t)}) \leq 2\gamma(\mathbf{g}, X)$. Furthermore, $e(\mathbf{h}^{(t)}) \geq e(\mathbf{g})/2$. From our initial choice of Q, we conclude that $\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle$ is (Q, B)-balanced for all such t.

(ii) *If* $m/2 \leq t \leq m - m^{5/8}$, *then* $\Pr\left(\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle \text{ is } (Q, B)\text{-balanced}\right) \geq 1 - m^{-\beta-1}/4$.
Recall that $\overline{H}^{(t)}$ can be viewed as a randomly chosen t-edge subgraph of G, or equivalently, its complement $H^{(t)}$ in G as a randomly chosen s-edge subgraph

of G, where $s = m - t$. Now we have

$$\gamma(\mathbf{h}^{(t)}, Y^{(t)}) = \max\left\{h^{(t)}_{\max} y^{(t)}_{\max}, {h^{(t)}_{\max}}^2\right\} \leq h^{(t)}_{\max}(g_{\max} + x_{\max}) \leq h^{(t)}_{\max} Qe(\mathbf{g})^{1/4}.$$

Hence $\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle$ will certainly be (Q, B)-balanced if $h^{(t)}_{\max} \leq Qe(\mathbf{h}^{(t)})^{1/2}/e(\mathbf{g})^{1/4}$, i.e., if the maximum vertex degree $h^{(t)}_{\max}$ of the random s-edge subgraph $H^{(t)}$ does not exceed $Qs^{1/2}/m^{1/4}$. We can estimate the probability of this event using Proposition 4.19 as follows: let $j \in [n]$ be any vertex with $g_j > 0$. Then if we colour green all g_j edges of G adjacent to j, and all other edges of G blue, the random variable $h^{(t)}_j$ is distributed as the number of green edges in a random sample (without replacement) of s edges of G. We are therefore in the situation of Proposition 4.19, with $Z = h^{(t)}_j$, $g = g_j$, and tail value $\alpha\mu = Qs^{1/2}/m^{1/4}$, where $\mu = sg_j/m$ is the mean of $h^{(t)}_j$. The factor α is quite large, viz.,

$$\alpha = \frac{Qm^{3/4}}{s^{1/2} g_j} \geq \frac{\sqrt{2} Qm^{1/4}}{g_{\max}} \geq 4\sqrt{2},$$

where we have used the facts that $s \leq m/2$ and $g_{\max} \leq (Q/4)m^{1/4}$. The tail value itself satisfies

$$\alpha\mu = \frac{Qs^{1/2}}{m^{1/4}} \geq Qm^{1/16},$$

since also $s \geq m^{5/8}$. Proposition 4.19 therefore yields

$$\Pr\left(h^{(t)}_j > \alpha\mu\right) \leq s\left(\frac{2\mathrm{e}}{\alpha}\right)^{\alpha\mu} \leq \frac{m}{2} c^{-m^{1/16}},$$

where $c = (2\sqrt{2}/\mathrm{e})^Q > 1$. Thus the probability that *any* vertex degree $h^{(t)}_j$ exceeds the bound is at most $m^2 c^{-m^{1/16}}$, which is less than $m^{-\beta-1}/4$ for all $m > B$, provided B is chosen large enough.

(iii) *If* $m - m^{5/8} \leq t \leq m - B$, *then* $\Pr\left(\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle \text{ is } (Q, B)\text{-balanced}\right) \geq 1 - m^{-\beta-1}/4$.

As in (ii) above, let $s = m - t$ and view $H^{(t)}$ as a randomly chosen s-edge subgraph of G. In view of (i), we may assume that $s \leq m/2$. By definition of γ, $\langle \mathbf{h}^{(t)}, Y^{(t)} \rangle$ will be (Q, B)-balanced if $h^{(t)}_{\max}$ and $y^{(t)}_{\max}$ are each bounded above by $Qe(\mathbf{h}^{(t)})^{1/4}$. In the case of $h^{(t)}_{\max}$ we proceed via Proposition 4.19 precisely as in (ii), only this time with tail value $\alpha\mu = Qs^{1/4}$. We find that

$$\alpha \geq \frac{Qm}{s^{3/4} g_{\max}} \geq 4\, m^{9/32},$$

since now $s \le m^{5/8}$. Further, $\alpha\mu = Qs^{1/4} \ge QB^{1/4}$, so we get the tail estimate

$$\Pr\left(h_j^{(t)} > \alpha\mu\right) \le s\left(\frac{\mathrm{e}}{2\,m^{9/32}}\right)^{QB^{1/4}}.$$

Thus the probability that any vertex degree $h_j^{(t)}$ exceeds $Qs^{1/4}$ is at most $m^2(cm)^{-\beta'}$, where $c > 0$ is fixed and β' can be made arbitrarily large by suitable choice of B. By setting B appropriately, we can clearly make this less than $m^{-\beta-1}/8$ for all $m > B$.

A similar argument can be used to handle $y_{\max}^{(t)}$: for a vertex $j \in [n]$ with $g_j > 0$, let $\Gamma(j)$ be the set of vertices adjacent to j in G. At this point we make use of the fact (refer to the definition of problem instance labels in the tree) that $Y^{(t)}$ includes only *essential* excluded edges, i.e., edges (i,k) for which both $h_i^{(t)} > 0$ and $h_k^{(t)} > 0$. From this it is clear that

$$y_j^{(t)} \le \left|\left\{i \in \Gamma(j) : h_i^{(t)} > 0\right\}\right|. \tag{4.10}$$

Now colour green all edges of G with an endpoint in $\Gamma(j)$, and the remainder blue, and again view $H^{(t)}$ as a random sample of size s from the edge set of G. Each time a green edge is selected, it contributes at most two to the right-hand side of (4.10). Thus $y_j^{(t)} \le 2Z$, where the random variable Z is the number of green edges in the sample, so the required tail probability may be estimated from Proposition 4.19 with $g = \sum_{i\in\Gamma(j)} g_i \le g_{\max}^2$, and $\alpha\mu = Qs^{1/4}/2$. The bounds on s in this range imply

$$\alpha \ge \frac{Qm}{2s^{3/4}g_{\max}^2} \ge \frac{8}{Q}m^{1/32},$$

and $\alpha\mu \ge QB^{1/4}/2$, so that

$$\Pr\left(y_j^{(t)} > 2\alpha\mu\right) \le \Pr\left(Z > \alpha\mu\right) \le s\left(\frac{\mathrm{e}\,Q}{4\,m^{1/32}}\right)^{QB^{1/4}/2}.$$

Exactly as above, this ensures that the probability that any vertex degree $y_j^{(t)}$ exceeds $Qs^{1/4}$ is at most $m^{-\beta-1}/8$ for all $m > B$, provided we make B large enough. Combining the bounds for $h_{\max}^{(t)}$ and $y_{\max}^{(t)}$, we arrive at (iii).

(iv) *If* $t \ge m - B$, *then* $\Pr\left(\langle \mathbf{h}^{(t)}, Y^{(t)}\rangle \text{ is } (Q,B)\text{-balanced}\right) = 1$.
This is true by definition, since $e(\mathbf{h}^{(t)}) = m - t \le B$.

In view of (i)–(iv), the probability of the conjunction in (4.9) is now easily seen to be bounded below by $1 - m^{-\beta}/4$, as claimed in the lemma. □

We conclude our discussion of graphs with specified degrees with some remarks on the counting problem. Recall again the procedure of Figure 1.3, which provides a way of estimating the number of leaves in a rooted tree given an almost uniform generator for leaves in the subtree rooted at any vertex. Consider the situation of Theorem 4.17: can we apply this technique to estimate the number of leaves in the pruned trees $T_R^{(Q,B)}(\mathrm{g}, X)$?

Note first that an almost uniform generator for the leaves in any maximal subtree S is available since we may simulate just this portion of the Markov chain $MC(\mathrm{g}, X)$, transitions out of S being censored. Moreover, the reduced chains clearly inherit the rapid mixing property, so the generator will be efficient provided only that the subtree has sufficiently many leaves. It is not hard to see that, by modifying slightly the method of selecting a subtree for the recursion, we can ensure that this condition always holds with high probability. Thus we get a f.p. randomised approximate counter for the leaves of such subtrees.

We may use this to approximate the number of leaves in $T_R^{(Q,B)}(\mathrm{g}, X)$ within ratio $1 + m^{-\beta}/3$ in polynomial time. But by Lemma 4.16 this number itself approximates $m!\,\#\mathrm{GRAPHS}(\mathrm{g}, X)$ within ratio $1 + m^{-\beta}/2$, so we are in fact able to approximate $\#\mathrm{GRAPHS}(\mathrm{g}, X)$ within ratio $1 + m^{-\beta}$ in polynomial time. We summarise this discussion in our final theorem.

Theorem 4.20 *For any fixed real β, there exists a polynomially time-bounded randomised approximate counter for* GRAPHS *within ratio $1 + e(\mathrm{g})^{-\beta}$, provided that the degrees are bounded as $\max\{g_{\max}, x_{\max}\} = \mathrm{O}\left(e(\mathrm{g})^{1/4}\right)$.* □

Theorem 4.20 implies the existence of polynomial time algorithmic methods for computing the number of labelled graphs with specified degrees (assuming that these are not too large) with a relative error which is smaller than any desired power of the number of edges. The asymptotic behaviour of such a counter thus compares very favourably with available analytic estimates, such as Theorem 4.13. While this is a remarkable theoretical result, we suspect that the various powers and constants accumulated in the reductions will render the method impractical if a high degree of accuracy is required.

Finally, we should observe that the counting problem for GRAPHS is apparently hard to solve *exactly* even under the degree restrictions imposed in this section, so that the approximation approach pursued here is justified.

Theorem 4.21 *The problem of evaluating* #GRAPHS *for instances* $\langle \mathrm{g}, X \rangle$ *whose degrees are bounded as* $\max\{g_{\max}, x_{\max}\} = \mathrm{O}\left(e(\mathrm{g})^{1/4}\right)$ *is #P-complete.*

Proof Note first that there is a simple reduction to the unrestricted version of this problem from the problem of counting perfect matchings in a graph G, under which the excluded graph X is the complement of G and the degree sequence is $(1, 1, \ldots, 1)$. The #P-completeness of the restricted version follows from the fact that the former problem remains #P-complete even for very dense graphs G, i.e., specifically when G has minimum vertex degree $n - \mathrm{O}(n^{1/4})$. This can be shown via the same reduction which is used to establish Theorem 3.9. □

Bibliography

[1] Aldous, D. Random walks on finite groups and rapidly mixing Markov chains. *Séminaire de Probabilités* XVII, 1981/82, Springer Lecture Notes in Mathematics 986, pp. 243–297

[2] Aldous, D. On the Markov chain simulation method for uniform combinatorial distributions and simulated annealing. *Probability in the Engineering and Informational Sciences* **1** (1987), pp. 33–46

[3] Aldous, D. and Diaconis, P. Shuffling cards and stopping times. *American Mathematical Monthly* **93** (1986), pp. 333–348

[4] Alon, N. Eigenvalues and expanders. *Combinatorica* **6** (1986), pp. 83–96

[5] Alon, N. and Milman, V.D. λ_1, isoperimetric inequalities for graphs and superconcentrators. *Journal of Combinatorial Theory Series B* **38** (1985), pp. 73–88

[6] Angluin, D. and Valiant, L.G. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Sciences* **18** (1979), pp. 155–193

[7] Bach, E. How to generate random integers with known factorisation. *Proceedings of the 15th ACM Symposium on the Theory of Computing,* 1983, pp. 184–188

[8] Bender, E.A. Asymptotic methods in enumeration. *SIAM Review* **16** (1974), pp. 485–515

[9] Berretti, A. and Sokal, A.D. New Monte Carlo method for the self-avoiding walk. *Journal of Statistical Physics* **40** (1985), pp. 483–531

[10] Binder, K. Monte Carlo investigations of phase transitions and critical phenomena. In *Phase Transitions and Critical Phenomena, Volume* 5b (C. Domb and M. S. Green eds.), Academic Press, London, 1976, pp. 1–105

[11] Bollobás, B. *Random Graphs* (Academic Press, London, 1985)

[12] Broder, A.Z. How hard is it to marry at random? (On the approximation of the permanent). *Proceedings of the 18th ACM Symposium on Theory of Computing,* 1986, pp. 50–58. Erratum in *Proceedings of the 20th ACM Symposium on Theory of Computing*, 1988, p. 551

[13] Broder, A.Z. and Shamir, E. On the second eigenvalue of random regular graphs. *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science,* 1987, pp. 286–294

[14] Cai, J.-y. and Hemachandra, L.A. Exact counting is as easy as approximate counting. Tehnical Report TR 86–761, Department of Computer Science, Cornell University (June 1986)

[15] Cheeger, J. A lower bound for the smallest eigenvalue of the Laplacian. In *Problems in Analysis* (R. C. Gunning ed.), Princeton University Press, New Jersey, 1970, pp. 195–199

[16] Dagum, P., Luby, M., Mihail, M. and Vazirani, U.V. Personal communication

[17] Diaconis, P. and Shahshahani, M. Generating a random permutation with random transpositions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **57** (1981), pp. 159–179

[18] Dixon, J.D. and Wilf, H.S. The random selection of unlabelled graphs. *Journal of Algorithms* **4** (1983), pp. 205–213

[19] Dodziuk, J. Difference equations, isoperimetric ineqality and transience of certain random walks. *Transactions of the American Mathematical Society* **284** (1984), pp. 787–794

[20] Edmonds, J. Paths, trees and flowers. *Canadian Journal of Mathematics* **17** (1965), pp. 449–467

[21] Edwards, K. The complexity of colouring problems on dense graphs. *Theoretical Computer Science* **43** (1986), pp. 337–343

[22] Erdős, P. and Graham, R.L. *Old and new problems and results in combinatorial number theory* (Monographie No. 28, L'Enseignement Mathématique, Geneva, 1980)

[23] Feller, W. *An introduction to probability theory and its applications, Volume* I (3rd ed.) (John Wiley, New York, 1968)

[24] Garey, M.R. and Johnson, D.S. *Computers and intractability* (Freeman, New York, 1979)

[25] Gill, J. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing* **6** (1977), pp. 675–695

[26] Guénoche, A. Random Spanning Tree. *Journal of Algorithms* **4** (1983), pp. 214–220

[27] Harary, F. and Palmer, E.M. *Graphical Enumeration* (Academic Press, New York, 1973)

[28] Heilmann, O.J. and Lieb, E.H. Theory of monomer-dimer systems. *Communications in Mathematical Physics* **25** (1972), pp. 190–232

[29] Hickey, T. and Cohen, J. Uniform random generation of strings in a context-free language. *SIAM Journal on Computing* **12** (1983), pp. 645–655

[30] Hopcroft, J.E. and Ullman, J.D. *Introduction to Automata Theory, Languages and Computation* (Addison-Wesley, Reading MA, 1979)

[31] Jerrum, M.R. 2-dimensional monomer-dimer systems are computationally intractable. *Journal of Statistical Physics* **48** (1987), pp. 121–134

[32] Jerrum, M.R. and Sinclair, A.J. Conductance and the rapid mixing property for Markov chains: the approximation of the permanent resolved. *Proceedings of the 20th ACM Symposium on Theory of Computing,* 1988, pp. 235–244

[33] Jerrum, M.R., Valiant, L.G. and Vazirani, V.V. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* **43** (1986), pp. 169–188

[34] Karp, R.M. and Luby, M. Monte-Carlo algorithms for enumeration and reliability problems. *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science,* 1983, pp. 56–64

[35] Kastelyn, P.W. Graph theory and crystal physics. In *Graph Theory and Theoretical Physics* (F. Harary ed.), Academic Press, London, 1967, pp. 43–110

[36] Keilson, J. *Markov chain models — rarity and exponentiality* (Springer-Verlag, New York, 1979)

[37] Kirkpatrick, S., Gellatt, C.D. and Vecchi, M.P. Optimisation by simulated annealing. *Science* **220** (May 1983), pp. 671–680

[38] Lautemann, C. On two computational models for probabilistic algorithms. Bericht-Nr. 82-15, Technische Universität Berlin, Fachbereich Informatik (November 1982)

[39] Lovász, L. *Combinatorial Problems and Exercises* (North Holland, Amsterdam, 1979)

[40] McKay, B.D. Asymptotics for symmetric 0-1 matrices with prescribed row sums. *Ars Combinatoria* **19A** (1985), pp. 15–25

[41] Meyer, A.R. and Paterson, M.S. With what frequency are apparently intractable problems difficult? Report MIT/LCS/TM–126, Laboratory for Computer Science, M.I.T. (February 1979)

[42] Mihail, M. The approximation of the permanent is still open. Preprint, Aiken Computation Laboratory, Harvard University, 1987

[43] Minc, H. *Permanents* (Addison-Wesley, Reading MA, 1978)

[44] Nijenhuis, A. and Wilf, H.S. *Combinatorial Algorithms* (2nd ed.) (Academic Press, Orlando, 1978)

[45] Nijenhuis, A. and Wilf, H.S. The enumeration of connected graphs and linked diagrams. *Journal of Combinatorial Theory Series A* **27** (1979), pp. 356–359

[46] Peleg, D. and Upfal, E. The token distribution problem. *Proceedings of the 27th Symposium on Foundations of Computer Science,* 1987, pp. 418–427

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

[48] Schnorr, C.P. Optimal algorithms for self-reducible problems. *Proceedings of the 3rd International Colloquium on Automata, Languages and Programming,* 1976, pp. 322–337

[49] Schnorr, C.P. On self-transformable combinatorial problems. *Mathematical Programming Study* **14** (1981), pp. 225–243

[50] Seneta, E. *Non-negative matrices and Markov chains* (2nd ed.) (Springer-Verlag, New York, 1973)

[51] Sinclair, A.J. and Jerrum, M.R. Approximate counting, uniform generation and rapidly mixing Markov chains. Internal Report CSR-241-87, Department of Computer Science, University of Edinburgh, October 1987. (Submitted to *Information and Computation*)

[52] Sipser, M. A complexity-theoretic approach to randomness. *Proceedings of the 15th ACM Symposium of Theory of Computing,* 1983, pp. 330–335

[53] Stockmeyer, L. The polynomial-time hierarchy. *Theoretical Computer Science* **3** (1977), pp. 1–22

[54] Stockmeyer, L. The complexity of approximate counting. *Proceedings of the 15th ACM Symposium on Theory of Computing,* 1983, pp. 118–126

[55] Tinhofer, G. On the generation of random graphs with given properties and known distribution. *Appl. Comput. Sci., Ber. Prakt. Inf.* **13** (1979), pp. 265–297

[56] Ullman, J.D. *Computational Aspects of VLSI* (Computer Science Press, Rockville, 1984)

[57] Valiant, L.G. The complexity of combinatorial computations: an introduction. *GI 8. Jahrestagung Informatik, Fachberichte Band 18* (1978), pp. 326–337

[58] Valiant, L.G. The complexity of computing the permanent. *Theoretical Computer Science* **8** (1979), pp. 189–201

[59] Valiant, L.G. The complexity of enumeration and reliability problems. *SIAM Journal on Computing* **8** (1979), pp. 410–421

[60] Welsh, D.J.A. Randomised algorithms. *Discrete Applied Mathematics* **5** (1983), pp. 133–145

[61] Wilf, H.S. The uniform selection of free trees. *Journal of Algorithms* **2** (1981), pp. 204–207

[62] Wormald, N.C. Generating random regular graphs. *Journal of Algorithms* **5** (1984), pp. 247–280

[63] Wormald, N.C. Generating random unlabelled graphs. *SIAM Journal on Computing* **16** (1987), pp. 717–727