Approximating the permanent is in RNC (Extended Abstract) *

J.Diaz† M.Serna† P.Spirakis‡

Abstract

In this work we look into the parallelization (in the NC sense) of the Markov Chain approach to almost random generation (as described in [Sin93]). We give a RNC almost uniform generator for perfect matchings in dense bipartite graphs, and its RNC reduction to its counting problem. As a consecuence we can classify the problem of getting an approximation scheme to the permanet of a boolean matrix in RNC. We also state other problems for which the same technique applies.

1 Introduction

Many problems involving the counting of the number of solutions of combinatorial structures, are well known to be difficult. Valiant defined the class $\#P$ of computationally equivalent counting problems [Val79b]. For many of the problems in this class, their decision counterpart is P. It is well known that unless the polynomial hierarchy collapses, $P \neq \#P$. This fact implies that for any $\#P$-complete problem, exact counting is apparently intractable for many interesting problems that are complete for $\#P$ [Pap94]. The most paradigmatic of these problems is to compute the permanent of a matrix, that turns out to be equivalent to count the number of perfect matchings in a dense bipartite graph [Val79a]. The hardness of these counting problems has led to approximate the counting. Pioneer work in this line was

*This research was supported by the ESPRIT BRA Program of the EC under contract no. 7141 ,project ALCOMII.

†Departament de Llenguatges i Sistemes, Universitat Politècnica Catalunya, Pau Gargallo 5, 08028-Barcelona {diaz,mjserna}@lsi.upc.es

‡CTI,Patras, Greece, spirakis@cti.gr

the paper [KLM89] where they construct a Randomized Fully Approximation Scheme for some difficult counting problems. Later, it was discovered that for the problems which have certain structural property, approximate counting is equivalent to almost uniform generation [JVV86]. The almost uniform generation problem consist in picking at random an element of a finite set characterized by some property, with a relative error of at most ϵ with respect to the probability that a given element is chosen.

Since for most of the interesting problems the approximation of counting is reduced to the almost uniform generation, we concentrate our efforts on the uniform generation problem. A technique that has proved to be very useful to solve the almost uniform generation problem, is the Markov Chain technique. Given a problem, define a Markov chain where the states are all possible solution, plus possibly a small fraction of quasi-solutions, and the transitions are certain probabilistics rules that allow us to remain in the same state or to pass to a new state. Under certain properties of the underlying graph, it can be proved that a polynomial (in the input size) random walk on the states give us an almost random generated element from the stationary distribution of the Chain. The difficulty of this method is to prove rapid convergence to the stationary distribution, what is called the "rapid mixing" property. Over the past years, a large body of literature has been devoted to the subject of almost random generation through Markov Chain, and methods of proving rapid mixing. Excellent surveys are [Sin93, Vaz91, Kan94].

A question of interest is the possibility of parallelizing the almost uniform generation and approximate counting problems. In other words, for a rapid-mixing Markov chain, given a node M we want to find a randomized parallel algorithm that is equivalent to some random walk on the chain, using polylog steps and a polynomial number of processors, without changing the distribution of outputs. We will refer such a process as a *RNC simulation*.

Consider the Markov chain used by Broder [Bro86] and by Jerrum and Sinclair [JS90] to generate almost uniformly perfect matchings in dense graphs. Teng has proved that the problem of given a sequential walk starting from a node M comptute the final node M' on the walk is P-complete [Ten95]. This result does not avoid the possibility of generating in parallel a almost uniform perfect matching.

In order to be able to obtain a RNC simulation, some assumptions are needed. First the random choices along the walk can be done independently for each state. That means that all random choices can be done in $\Theta(1)$ parallel steps, at the beginning. Second, at least one state of the Markov

Chain can be obtained in NC (or RNC).

In the present work, we introduce a new Markov chain, that for dense bipartite graphs generates perfect matchings and near-perfect matchings with high probability, this chain has a stationary distribution with special properties that we use to obtain almost uniform generation of perfect matchings. This Markovian process is RNC-simulated, to get a RNC almost uniform generator for perfect matchings. We obtain similar results for the spanning tree generation problem, directly from the Markov chain given by Sinclair [Sin93]. We extend the perfect matching generation to the counting problem, and we show that the problem of approximating the permanent of a dense boolean matrix is in RNC.

2 Almost uniform generation of perfect matchings in dense bipartite graphs is in RNC

Let $G = (V_1, V_2, E)$ be a dense bipartite graph, that is every vertex is connected to at least half of the vertices, $|V_1| = |V_2| = n$ and for $k \in [n]$ let $M_k(G)$ denote the set of matchings of size k in G. It is well known that testing if G has a perfect matching is in RNC [MVV87].

We begin by defining the merging operation of of a matching $M_k(G)$ with an edge e of G, the operation is defined in terms of some probability s to be fixed later. This merging operation will produce a new (partial) matching. Define the merging $M_k \mu e$ by the following procedure;

Merge simple

(1) Construct connected components of the bipartite graph induced by $M_k \cup \{e\}$.

(2) If there is a connected component of length 3,

(2.1) with probability s remove the edge e (we are left with two connected components each with one edge),

(2.2) with probability $1 - s$ choose to remove one of the edges in M_k (to get one connected component with 2 edges, case 3).

(3) If there is a connected component with two edges, with probability $1/2$ choose one of the edges to be removed.

(4) The remaining edges are the output of $M_k \mu e$.

Given a bipartite dense graph G with $2n$ nodes, consider the Markov Chain whose states are all the matchings $M_n \cup M_{n-1} \cup \cdots \cup M_1$ and from any state M the transitions are defined by getting with probability $1/|E|$ an edge e of G and doing the merging $M\mu e$.

This Markov chain differs from the one used by Broder and Jerrum-Sinclair. There are more states and the transitions are defined in a different way. Notice that for any $n \leq k \leq 3$ and any $i \geq 2$ the probability of going in one step from M_k to M_{k-i} is 0.

Theorem 1 *The Markov chain is ergodic, Let $\Pi_0(A)$ and $\Pi_0(B)$ be the steady state probabilities of any two $A, B \in M_n(G)$, and let $\Pi_1(A)$ and $\Pi_1(B)$ be the steady state probabilities of any two $A, B \in M_{n-1}(G)$. The the Markov chain satisfies:*

For all $A, B \in M_n(G)$ we have $|\Pi_0(A) - \Pi_0(B)| = \Theta(1/n)$,
for all $A, B \in M_{n-1}(G)$ we have $|\Pi_1(A) - \Pi_1(B)| = \Theta(1/n)$.

Thus it is almost uniform with respect to perfect matching and near-perfect matchings. Furthermore for every $A \in M_n(G)$ and $B \in M_{n-1}(G)$

$$\frac{\Pi_1(B)}{\Pi_0(A)} = 1 - (\frac{n-2}{2m} + \frac{m-n+1}{m}s) + \Theta(1/n).$$

Moreover the chain has the rapid mixing property.

Proof. Ergodicity follows from the fact that after a finite number of steps there is a positive probability of reaching any state from any other.

Conside first the simple chain C_1 of Figure 2. We have $\Pi_0 = \Pi_1 = 1/2$. Consider for some $\epsilon > 0$ its ϵ-perturbation C_1' (see Figure 3). Let Π_i' be the $\Pr(S_i)$ in the stationary distribution, from the equation $\Pi = \Pi.P$ we get,

$$\begin{aligned}
\Pi_0' &= \Pi_0'/2 + \Pi_1'/2 \\
\Pi_1' &= \Pi_0'/2 + \Pi_1'(1/2 - \epsilon) + \Pi_2'/2 \\
\Pi_2' &= \Pi_1'\epsilon + \Pi_2'/2.
\end{aligned}$$

which implies $(\Pi_0', \Pi_1', \Pi_2') = (1/2(1+\epsilon), 1/2(1+\epsilon), \epsilon/1+\epsilon)$ and
$|\Pi_0 - \Pi_0'| = |\Pi_1 - \Pi_1'| \leq \epsilon/2$.

Consider now the multi-stage ϵ-perturbation $\hat{C}_1$ of C_1, (see Figure 4). We get $\hat{\Pi}_{n-1} = \hat{\Pi}_{n-1}/2 + \hat{\Pi}_{n-2}\epsilon$, that means $\hat{\Pi}_{n-1} = 2\hat{\Pi}_{n-2}\epsilon$.

Similarly we get $\hat{\Pi}_k = \Theta(\epsilon^{k-2})2\hat{\Pi}_2$ and $\hat{\Pi}_0 = \hat{\Pi}_0(1/2-\epsilon) + \hat{\Pi}_1/2$ from which we get $\hat{\Pi}_1^2 = \frac{2\epsilon}{(1+2\epsilon)^2}\hat{\Pi}_1$ and $\hat{\Pi}_0 = \hat{\Pi}_1\frac{1}{7}1+\epsilon$. Since $\sum \hat{\Pi}_i = 1$ we obtain

$$\hat{\Pi}_1\left(\frac{1}{1+2\epsilon} + 1 + \frac{2\epsilon}{(1+2\epsilon)^2} + \Theta(\epsilon^2) + \cdots + \Theta(\epsilon^{n-\epsilon})\right) = 1$$

therefore for small ϵ we have $\hat{\Pi}_1 = 1/2 + \Theta(\epsilon)$.

Hence, $|\hat{\Pi}_1 - \Pi_1| = \Theta(\epsilon)$ and $|\hat{\Pi}_0 - \Pi_0| = \Theta(\epsilon)$

It is easy to see that the markov chain with $s = 0$, call it H, produces uniformly $M \in M_n \cup M_{n-1}$. Let us prove that the chain H_s defined by the simple merge is an $(1-s)$-perturbation of H. Consider the stationary vector distribution on $[\Pi, \Pi, \ldots, \Pi, 0, \ldots, 0]$ the 0's corresponding to matchings M_k with $k < n-1$. The Π are a solution to the equation $\Pi = \Pi\, P(H)$ with $\sum \Pi_i = 1$.

The chain H_s perturbs some transition probabilities p_{ij} downwords by changing them from 0 to ϵ, thus for every $M \in M_n \cup M_{n-1}$ we get the relation $|\Pi_\epsilon(n) - \Pi(n)| = \Theta(\epsilon)$. Considering $\epsilon = 1 - s = 2/n = \Theta(1/n)$ we get that for every $M \in M_n \cup M_{n-1}$, $|\Pi_H(M) - \Pi_{H_S}(M)| = \Theta(1/n)$, proving the second part of Theorem 1.

Consider now the chain H'_S which is like H but for all $M \in M_n$ has the same $\Pi'_{H'_S}(M)$; and for all $M \in M_{n-1}$ it also has the same $\Pi'_{H'_S}(M)$. For such H'_S we can prove in a similar way as before, that for every $M \in M_n \cup M_{n-1}$ we get

$$|\Pi_{H_S}(M) - \Pi_{H'_S}(M)| = \Theta(1/n).$$

But we know that in H'_S, $\frac{\Pi'_1}{\Pi'_0} = 1 - (\frac{n-2}{m} + s\frac{m-n-1}{m})$. Thus in H_s for all $M \in M_n(G)$ and $M' \in M_{n-1}(G)$ we have $\frac{\Pi(M')}{\Pi(M)} = 1 - \left(\frac{n-2}{m} + s\frac{m-n-1}{m}\right) + \Theta(1/n)$ proving the third part of the theorem.

To prove rapid mixing, define a canonical path argument as in Jerrum Sinclair [JS90, Sin93]: Given matchings $M, M' \in M_k(G)$ such that its symmetric difference consists of paths with even length, the canonical path will be the shortest path using operation (3). In the general case, given M and M', there is a unique pair M_1, M'_1 from the previous case and such that M_1 can be obtained from M by elimination of edges (2.1) and from M'_1 we can obtain M' by addition of edges. Define the canonical path as the path that minimizes the number of transitions from M to M_1, then from M_1 to M'_1 and finally from M'_1 to M'. Then using the techniques as in [Sin93], it is easy to bound the conductance by $1/\Theta(n^5)$. $\square$

To simulate in parallel a walk on this chain, we need to extend the merge definition for the merging of two matchings using a sequence $\vec{e}$ of edges of G with polynomial length.

Given a partial matching M_1 and a sequence of its edges $\vec{e}_1$ and given another partial matching M_2 and a sequence of its edges $\vec{e}_2$ we define a merging operation with respect to a sequence $M_1 \mu_{\vec{e}} M_2$, as the matching returned by the following randomized process,

Merge $\vec{e}$

Let $\vec{a}$ be a sequence obtained by concatenation of $\vec{e}_1$ and $\vec{e}_2$ plus the deletion of repeated elements in $\vec{e}_2$

(1) Construct connected components of the bipartite graph induced by $M_1 \cup M_2$.

(2) For each connected component with an odd number of edges, do,

- (2.1) Compute the size i of the largest matching in the component
- (2.2) With probability s^{i-1} keep the bigger matching,
- (2.3) With probability $1 - s^{i-1}$ remove the end edge of the chain that appears last in the sequence $\vec{a}$,

(3) For each actual component with an even number of edges, with probability $1/2$ choose to keep edges in M_1 or in M_2.

(4) The remaining edges are the output of $M_1 \mu_{\vec{e}} M_2$. The new sequence will be $\vec{s}$ after deleting the edges that do not appear in $M_1 \mu_{\vec{e}} M_2$.

Notice that for independent components the probability of applying step 2 or step 3 is the product of their probabilities. The following result is straightforward to proof,

Lemma 1 *The merge operation can be implemented in RNC.*

Theorem 2 *Let M_1, M_2, M_3 be (partial) matchings and G' the graph $M_1 \cup M_2 \cup M_3$. Let p_1 (p_2) be the probability of obtaining a perfect matching (a near-perfect matching) when merging in some order. Then $\mu_{\vec{e}}$ is associative wrt p_1, p_2 i.e. for $i = 1, 2$ $p_i(M_1 \mu_{\vec{e}}(M_2 \mu_{\vec{e}} M_3)) = p_i((M_1 \mu_{\vec{e}} M_2)\mu_{\vec{e}} M_3)$*

The proof is done by induction on the number of edges b_2 in M_2 and b_3 in M_3. The base case is when $b_2 = b_3 = 1$, and in this case the proof follows the examination of all possible cases. ¿From the base case, it is easy to derive the inductive step, by realizing that a matching can be obtained by merging its edges.

Now, consider a sequence $\vec{e} = e_1, \ldots, e_k$ of edges of G randomly sampled, and an initial perfect matching M. Let J_1 denote the sequential merge $(\cdots(M\mu_{\vec{e}}e_1)\ldots)\mu_{\vec{e}}e_{p(n)})$, and let J_2 denote the merge performed in parallel as shown in Figure 1 in the Appendix. Then for $i = 1, 2$ we have $p_i(J_1) = p_i(J_2)$. Let $\vec{e}$ be partitioned into subsequences $\{S_i\}_{i=1}^{\lambda}$ such that no edge appears twice in any subsequence. Then,

Theorem 3 *Let G be a dense bipartite graph, and let $k \geq p(n)$ for a suitable selected polynomial $p(n)$. Then*

$$p_1(J_2) + p_2(J_2) > c$$

for some constant c independent of n.

Proof. Let $M_{\vec{e}}$ be the partial matching obtained after merging in parallel all edges in $\vec{e}$ (i.e. at the root of tree T in Fig. 1) and F be the number of free nodes in $M\mu_{\vec{e}}M_{\vec{e}}$. Since M is a perfect matching the maximum number of free nodes created by $M\mu_{\vec{e}}M_{\vec{e}}$ will happen when each one of the edges in $M_{\vec{e}}$ form a separate odd components. Let $m_{\vec{e}}$ be the number of edges in $M_{\vec{e}}$. We can form at most $\min\{m_{\vec{e}}, n - m_{\vec{e}}\} \leq n/2$ odd components , leading to an average number of free nodes $\overline{F} \leq 2(1-s)n/2$. To get $\overline{F} < 1$ we need to choose $s > 1 - 1/n$. Since $F > 0$, using Markov inequality we get $\Pr(F \geq 2) \leq \overline{F}/2 \leq 1/2$. □

Using the merge described above, we describe the following RNC simulation of the above Markov chain,

Statistical simulation

(1) Compute in RNC a perfect matching M,

(2) In parallel select random and independently a polynomial $p(n)$ number of edges $\vec{e} = e_1, \ldots, e_{p(n)}$. Uniformly at random choose an order $\vec{\sigma}$ of the edges in M.

(3) In parallel partition the edge sequence in segments with no repetitions of edges.

(4) In parallel for each edge segment construct a matching by a balanced tree of $\mu_{\vec{e}}$ operations with the edges of the segment as leaves.

(5) We use a balanced tree of $\mu_{\vec{e}}$ operations with the matching obtained in the previous step as leaves. This produces a matching M'.

(6) Compute $M_f = M\mu_{\vec{e}}M'$, using $\vec{\sigma}$ as ordered sequence.

(7) If M_f is not a perfect or a near-perfect matching go to (2).

Theorem 4 *The statistical simulation is a RNC-simulation of the Markov chain defined previously. Furthermore it outputs and element in $M_n \cup M_{n-1}$, with probability at least $1 - n^{-\alpha}$ for some constant α*

Proof. All steps above can be done in parallel using as maximun $O(\log^2 n)$ parallel steps and a suffiently large polynomial number of processors. Moreover using Lemma 3, we can bound the error probability by $1 - n^{-\alpha}$, where α is a constant independent of n. □

As a corollary of Theorem 1, Lemma 1 and the fact that the graph is dense, we get the following result,

Theorem 5 *Given a dense bipartite graph G, there is an almost uniform RNC-generator for M_n.*

3 Approximating the permanent

In this section we are going to show that the problem of approximating the permanent is in RNC. Recall that given a boolean matrix $A = [a_{ij}]$ the permanent of A is defined by per $(A) = \sum_\sigma \prod_{i=0}^{n-1} a_{i\sigma(i)}$ where the sum is over all permutations σ on $[n]$. In the previous section we have proved that the problem of generating almost uniformly a perfect matching in bipartite graphs is in RNC. Here we adapt the reduction from almost uniform generation of perfect matchings to approximate counting used in [Bro86] to put in RNC the problem of approximating the number of perfect matchings.

The following result is adapted from Lemma 2 in [Bro86],

Lemma 2 *Assume we are given RNC-approximation schemes to compute the ratios $\{|M_k(G)|/|M_{k+1}(G)|\}_{k=1}^{n-1}$, then we have a NC construction of a RNC-approximation scheme for the ratio $|M_1(G)|/|M_n(G)|$.*

Let ρ be the approximation of the ratio Π_1/Π_0, given in Theorem 1, we consider the following sampling procedure:

Sampling procedure

(1) Execute N times in parallel the Statistical Simulation to get a sample $S = \{S_1, \ldots S_N\}$.

(2) Select each $S_i \in M_n(G)$ with probability $1/1+\rho$.

(3) Select each $S_i \in M_{n-1}(G)$ with probability $\rho/1+\rho$.

(4) Select each $S_i \in M_k(G)$ for $k < n-1$ with probability zero.

(5) Let S' the selected elements in steps 1 to 4. Compute the number q of elements from M_n in S' and the number r of elements from M_{n-1} in S'.

(6) Output r/q.

Using Theorems 1 and 4, and elementary probability theory (Chernoff's bounds) we can show that the output of the sampling procedure is a precise estimator of $|M_{n-1}(G)|/|M_n(G)|$, thus we get,

Theorem 6 *Given a dense bipartite graph G, and an ϵ we can compute in RNC an ϵ–approximation to the ratio $|M_{n-1}(G)|/|M_n(G)|$.*

Given a bipartite graph $G = (V_1, V_2, E)$, $|V_1| = |V_2| = n$ and any $k \in [n]$, let $G_k = (V_1', V_2', E'$ where $V_1' = (V_1 \cup \{x_1, \ldots, x_k\}, V_2 \cup \{y_1, \ldots, y_k\}, E \cup \{x_1, \ldots, x_k\} \times V_2 \cup V_1 \times \{y_1, \ldots, y_k\})$. For each G and k, the construction of G_k can be done in NC. As in [Bro86]

$$\frac{|M_{n+k}(G_k)|}{|M_{n+k-1}(G_k)|} = \frac{|M_{n-k}(G)|}{2k|M_{n-k}(G)| + |M_{n-k+1}(G)| + (k+1)^2|M_{n-k-1}(G)|}$$

from the previous equation, if we start from a given evaluation of the ratio $|M_{n+k}(G_k)|/|M_{n+k-1}(G_k)|$ it is possible to obtain an estimation of $|M_{n-k}(G)|/|M_{n-k+1}(G)|$, moreover these computations can be done in NC, with a polynomial number of processors. We can run the random sampling procedure for G_k, in polylog parallel time, to obtain an estimation for $|M_{n+k}(G_k)|/|M_{n+k-1}(G_k)|$. Moreover this procedure can be executed in parallel for all values of k. Therefore using lemma 2 we get,

Theorem 7 *Given a dense bipartite graph $G = (V_1, V_2, E)$ with $|V_1| = |V_2| = n$ there exist a RNC-approximation scheme to the number of perfect matchings in G.*

Corollary 1 *Given a boolean matrix A, to compute an approximation to its permanent is in RNC.*

Similar techniques like the ones developed in this section can be applied to other counting problems that have been approximated using the Markov approach. In particular it is easy to see that the approximations to the following counting problems can be computed in RNC:

1. Counting 2-factors in dense graphs [JS90],
2. Counting the number of Hammiltonian cycle in dense grafs [DFJ94],
3. Counting the number of Eulerian orientations in a graph [MP92].

Acknwoledgement: We thank Milena Mihail for her comments on an earlier draft.

References

[Bro86] A. Broder. How hard is to marry at random. In *18th ACM Symposium on Theory of Computing*, pages 50–58, 1986.

[Bro89] A. Broder. Generating random spanning trees. In *30th IEEE Symposium on Foundations of Computer Science*, pages 442–447, 1989.

[DFJ94] M. Dyer, A. Frieze, and M. Jerrum. Approximately counting Hamilton cycles in dense graphs. In *5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 336–343, 1994.

[Gue83] A. Guenoche. Random spanning tree. *Journal of Algorithms*, 4:214–220, 1983.

[JS90] M. Jerrum and A. Sinclair. Fast uniform generation of regular graphs. *Theoretical Computer Science*, 73:91–100, 1990.

[JVV86] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[Kan94] R.. Kannan. Markov chains and polynomial time algorithms. In *35th IEEE Symposium on Foundations of Computer Science*, pages 656–671, 1994.

[KLM89] R.M.. Karp, M. Luby, and N. Madras. Monte-carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 18:429–448, 1989.

[MP92] M. Mihail and Winkel. P. On the number of eulerian orientations of a graph. In *3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 526–531, 1992.

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is an easy as matrix inversion. In *19th. Annual ACM Symposium on Theory on Computing*, pages 355–365. ACM, 1987.

[Pap94] C. Papadimitriu. *Computational Complexity*. Addison-Wesley, 1994.

[Sin93] A. Sinclair. *Algorithm for random generation and counting: A Markov chain approach*. Birkhäuser, Boston, 1993.

[Ten95] S-H. Teng. Independent sets versus perfect matchings. *Theoretical Computer Science*, 1995.

[Val79a] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[Val79b] L.G. Valiant. The complexity of enumeration and reability problems. *SIAM J on Computing*, 8:410–421, 1979.

[Vaz91] V. Vazirani. Rapidly mixing Markov chains. In B. Bollobas, editor, *Probabilistic combinatorics and its applications*, pages 99–121. American Mathematical Society, 1991.

Figures

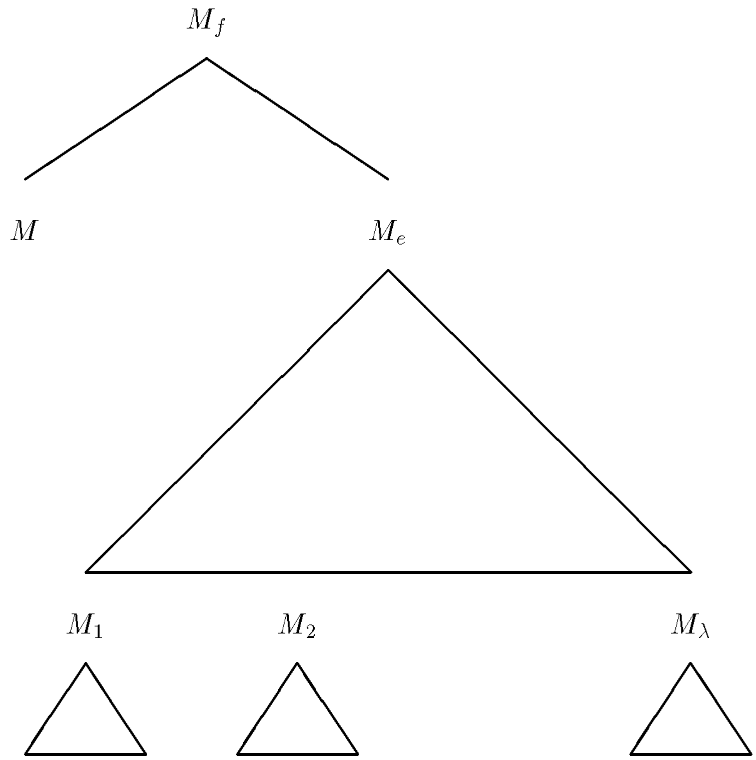

Figure 1: Tree of mergings

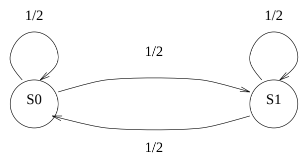

Figure 2: The simple chain C1

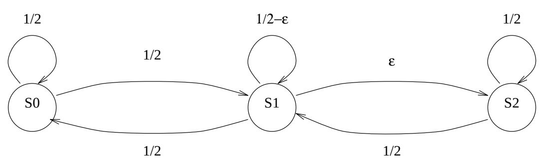

Figure 3: The single perturbation of C1

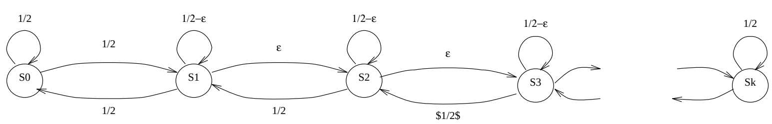

Figure 4: The multistage perturbation of C1