Sampling matchings in parallel *

J. Diaz† M. Serna† P. Spirakis‡

November 27, 1996

Abstract

We give a RNC algorithm to sample matchings from an almost uniform distribution on the set of matchings of all sizes in a graph. The technique used is based on the definition of a genetic system that converges to the uniform distribution. This genetic system is defined through several refinements of a rapidly mixing Markov chain. The parallel simulation of the genetic system gives a RNC almost uniform generator for the set of all matchings.

1 Introduction

Given an $\epsilon > 0$, the *almost uniform generation problem* consist in picking at random an element of a finite set according to some distribution Π, such that the variation distance between Π and the uniform distribution is at most ϵ. A technique that has proved to be very useful for solving the almost uniform generation problem, is the *Markov chain technique*. Given a problem, define a Markov chain where the set of states contain all possible solutions. Transitions are probabilistic rules that allow a move from state to state. Under certain properties of the underlying graph representing the Markov chain, it can be proved that a polynomial random walk on the states gives us an almost randomly generated element from the stationary distribution of the chain, the polynomial is on the size of the input, not on the size of the chain. For instance, a n-dimensional hypercube with the vertices vectors

*This research was partially supported by the ESPRIT Long Term Research Project no. 20244 – ALCOM-IT. The fourth author was also supported by the Centre Recerca Matemàtica, at Bellaterra.

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

‡Computer Technology Institute (C.T.I.), Patras, Greece, spirakis@cti.gr

in $\{0,1\}^n$, can be seen as the underlying graph of a Markov chain, where the states are all the 2^n bit vectors and from any state v the transitions are defined by flipping with uniform distribution one of the bits in v. To generate with almost uniform distribution a vector in $\{0,1\}^n$, proceed by doing a random walk, starting from any chosen vector. In [Sin93a] is proved that after a polynomial number of steps, the random walk will end up in a random vector with almost uniform distribution. This example is trivial and we include to illustrate the basic methodology.

The difficulty of this technique is to prove convergence in a polynomial number of steps to the stationary distribution, usually referred to as the *rapid mixing* property. Several methods have been used to prove this property, one of the most oftenly used bounds a topological parameter called the *conductance* using a canonical paths argument [JS89, Sin93b, Sin93a]. Over the past years, a large body of literature has been devoted to the subject of almost uniform generation through Markov chains and methods for proving rapid mixing. Excellent surveys can be found in [Sin93a, Vaz91, Kan94] and chapters 6 and 11 of [MR95].

A question of general interest is the possibility of parallelizing the almost uniform generation. In the case of of the n-dimensional hypercub it is easy to obtain a random parallel almost uniform generation. We can proceed as follows: In parallel, choose randomly l numbers in the range $1,\ldots,n$ that corresponds to the bits to be changed. As initial n-bit vector, take the 0 vector. Thus the problem can be stated as given a n-bit vector v together with a sequence of l integers $a_1,\ldots,a_l \in \{1,\ldots n\}$. Compute the n-bit vector $v^* = [v, < a_1,\ldots,a_l >]$. Where the basic operation is switch the i_i-th bit. In order to perform the computation in parallel, we change the representation, each integer v_i to be represented by a n-bit vector v_i in which all components are 0 except the i-th, that is 1. Notice the switch of bit i in vector w can be computed by $w \oplus v_i$ where $\oplus$ denotes the modulo two addition. The problem can be reformulated as $v^* = (\ldots(v \oplus v_1)\ldots) \oplus v_l$. As $\oplus$ is an associative operation we can apply the *tree contraction* technique [JaJ92] and compute v^* in NC. Therefore we can have a NC almost uniform generator for a vector in $\{0,1\}^n$.

In general things are not so easy. The direct NC simmulation of the random walk on the underlying Markov chain is in general P-complete. For instance, Teng has shown that given one states in the Markov chain used by Jerrum and Sinclair to approximate the Permanent, and a path to compute the state reached by the path is P-complete [Ten95]. In Section 5 of this manuscript, we show the same result for the Markov chain introduced in Section 2. To obtain the RNC generator instead of using a Markov chain,

we define a *genetic system*. In a genetic system, from a given initial distribution, new generations are grown by *mating* two randomly selected parents. Through this paper, the mating operators will produce only one offspring, thus our genetic system is non-quadratic, so it differs of the systems used by [RSW92] and [RRS95]. For instance, in the example of the n-dimensional hypercube, the mating operator could be defined as the direct sum of two bit vectors. The generations are new distributions over the set of elements produced by the mating operation. To analyze the *mixing time* we relate the genetic system with a sequence of Markov chains, the evolution of the second eigenvalue through the constructed sequence of Markov chains, gives the desired mixing time of $O(\log n)$.

To perform the simulation of the genetic system in RNC, we need a restricted size model in which each generation is replaced by a *population* of polynomial size. It is worth to remark that through the text, while by a *generation* we mean a distribution over a set, a *population* denotes a multiset produced as a sample of the corresponding generation.

In Section 2, we give the formal definition of the Markov chain to generate almost uniformly a matching in a given graph, define the genetic system, and prove convergence to the uniform distribution of the genetic system and the chains. In Section 3, we analyze the mixing time of the genetic system, proving a $O(\log n)$ convergence time. Section 4 gives the scheme to perform the simulation of the genetic system in parallel, showing that a polynomial size model is enough to carry on the simulation. In Section 5 we present the P-completeness of the Markov random walk for the chain presented in Section 2. Finally in Section 6 we give some open problems and conclusions.

2 The Markov chain and the genetic system

Given a graph $G = (V, E)$ with $|V| = n$ and $|E| = \mu$. For $k \in \{0, \cdots, \lfloor n/2 \rfloor\}$, let $M_k(G)$ denote the set of matchings of size k in G, and denote by $M = \cup_k M_k$ the set of all its matchings. From now on, G will denote the input graph. Recall that counting the total number of matchings in a given graph is known to be $\#P$-complete [Sin93a].

Let $\mathcal{M}$ be the Markov chain defined in [Vaz91] for the uniform generation of all matchings in a given graph G. The chain contains all elements of M as space state, and the transitions are defined in the following way:

Definition of transitions in $\mathcal{M}$:

Given a matching $m \in M$,

(0) Sample uniformly a random edge $e = (u, v)$,

(1) If e is in m then go to matching $m - \{e\}$

(2) If $m \cup \{e\}$ is a matching, then go to new matching $m \cup \{e\}$

(3) Otherwise stay in m.

The following result is well known (see for example [Vaz91])

Theorem 1. *The Markov chain $\mathcal{M}$ converges to the uniform distribution of all matchings in G and it is rapidly mixing.*

In Section 5 we show that from a given state in this chain, and a sequence of selected edges, we can not obtain in NC the final state reached, unless P=NC.

Let us consider a probability distribution Π on the set M of all matchings in G. Define the Markov chain $\mathcal{M}(\Pi)$ as a modification of the previous chain $\mathcal{M}$. The chain $\mathcal{M}(\Pi)$ will have the same state space as $\mathcal{M}$ and a transition in $\mathcal{M}(\Pi)$ will be defined as follows:

Definition of transitions in $\mathcal{M}(\Pi)$:

Given a matching $m \in M$,

(0) Sample a matching m_i according to distribution Π,

(1) Sort randomly the edges of m_i.

(2) From state m, go to the state m_k resulting of following in $\mathcal{M}$ the path defined by ordered edges of m_i.

The Markov chain $\mathcal{M}(\Pi)$ is a generalization of the chain $\mathcal{M}$. Furthermore, in the particular case that the distribution Π assigns uniform probability to the set of matchings with one edge, and assigns probability 0 to the remaining matchings, the chain $\mathcal{M}(\Pi)$ coincides with $\mathcal{M}$.

Let $\Pi(j)$ denote the probability of choosing m_j from the distribution Π, and $\Pi(e)$ be the probability of choosing a matching containing edge e. We can move in $\mathcal{M}(\Pi)$ from any state to any other state in at most two steps. Therefore provided the distribution Π assigns positive probability to all edges in G we can move in $\mathcal{M}(\Pi)$ from any state to any other state in at most n steps. Therefore, the probability of reaching any matching in at most n steps is greater than 0. Thus we get,

Lemma 1. *Let Π be a probability distribution such that for every edge e we have $\Pi(e) > 0$, then $\mathcal{M}(\Pi)$ is ergodic.*

Given three matchings m_i, m_j and m_k, let $P(i,j,k)$ denote the probability of going from m_i to m_k following a sequence given by the edges of m_j. In $\mathcal{M}(\Pi)$ the resulting matching is the same independently of the order of the edges, thus $P(i,j,k)$ is either 1 or 0, with $\sum_k P(i,j,k) = 1$. The symmetry of $\mathcal{M}$ implies that if we can go from m_i to m_k following a sequence given by m_j, then with the same probability we can go from m_k to m_i following the reverse sequence, therefore $P(i,j,k) = P(k,j,i)$.

The i,k coefficient $\Pi(i,k)$ in the transition matrix of $\mathcal{M}(\Pi)$ is given by $\Pi(i,k) = \sum_{m_j \in M} P(i,j,k) \cdot \Pi(j)$. Moreover, as $P(i,j,k) = P(k,j,i)$ we get that $\Pi(i,k) = \Pi(k,i)$. Thus,

Corollary 1. *The chain $\mathcal{M}(\Pi)$ is symmetric, for any distribution Π.*

We define a *genetic system* $\mathcal{G}$ over the population of all matchings M that will produce the next generation according to a mating rule based in the transitions of $\mathcal{M}(\Pi)$.

Definition 1 (Mating Rule). *From parents m_l and m_r, sort randomly the edges of m_r. The offspring m_k is the matching resulting of the walk in $\mathcal{M}$ starting from state m_l and following the path defined by ordered edges of m_r.*

To define a system evolving in time t, start from a given initial generation Π_0 over M at $t = 0$. The generation at time $t+1$ is obtained from the generation Π_t at time t, by sampling two matchings m_l and m_r according to Π_t, and applying the mating rule to m_l and m_r. The system evolves according to the following dynamical equation,

$$\Pi_{t+1}(k) = \sum_{m_l \in M} \Pi_t(l) \cdot \sum_{m_r \in M} P(l,r,k) \cdot \Pi_t(r) \tag{1}$$

Theorem 2. *For any distribution Π, the system $\mathcal{G}$ and the chain $\mathcal{M}(\Pi)$ have the uniform distribution as fix point.*

Proof. Let us consider the equation

$$\Pi_{t+1}(k) = \sum_{m_l} \Pi_t(l) \cdot \sum m_r P(l,r,k) \cdot \Pi_\alpha(r).$$

Notice that with $\alpha = t$ we have the genetic system $\mathcal{G}$, and with $\Pi_\alpha = \Pi$ we have the chain $\mathcal{M}(\Pi)$. Then as $P(l,r,k) = P(k,r,l)$ we get

$$\sum_{m_l} \Pi_t(l) \cdot \sum_{m_r} P(l,r,k) \cdot \Pi_\alpha(r) = \sum_{m_l} \Pi_t(l) \cdot \sum_{m_r} P(k,r,l) \cdot \Pi_\alpha(r).$$

We must prove that if at some time t we get the uniform distribution $\Pi_t(l) = 1/\beta$, where β is the total number of matchings in G, we also get $\Pi_{t+1}(l) = 1/\beta$. Substituting the value of Π_t in the above equation,

$$\begin{aligned}
\Pi_{t+1}(k) &= \sum_{m_l} \frac{1}{\beta} \cdot \sum_{m_r} P(k,r,l) \cdot \Pi_\alpha(r) \\
&= \frac{1}{\beta} \cdot \sum_{m_l} \sum_{m_r} P(k,r,l) \cdot \Pi_\alpha(r) \\
&= \frac{1}{\beta} \cdot \sum_{m_r} \Pi_\alpha(r) \cdot \sum_{m_l} P(k,r,l) \\
&= \frac{1}{\beta} \cdot \sum_{m_r} \Pi_\alpha(r) = \frac{1}{\beta}.
\end{aligned}$$

□

The previous proof is independent of the choice of distribution. Therefore by Lemma 1, Theorem 2, and classical Markov chain theory we can conclude

Theorem 3. *If for every edge e we have $\Pi(e) > 0$, then the chain $\mathcal{M}(\Pi)$ converges to the uniform distribution on the set of all matchings M.*

Let us turn to prove that the genetic system $\mathcal{G}$ also converges to the uniform distribution,

Theorem 4. *If for every edge e, we have $\Pi_0(e) > 0$, then the genetic system $\mathcal{G}$ converges to the uniform distribution on the set M.*

Proof. Let us recall that obtaining generation $i+1$ in the genetic system $\mathcal{G}$ can be seen as one step in the Markov chain $\mathcal{M}(\Pi_i)$; in other words, assuming that $A(\Pi_i)$ is the transition matrix of the chain $\mathcal{M}(\Pi_i)$ then $\Pi_{i+1} = \Pi_i \cdot A(\Pi_i)$. Notice that the hypothesis condition on the initial distribution Π_0 assures the ergodicity of $\mathcal{M}(\Pi_i)$, and by Theorem 3 the convergence of $\mathcal{M}(\Pi_i)$ to the uniform distribution. Using standard Markov chain techniques (see for ex. [Ry70]), for every i we get $||\Pi_{i+1} - \Pi_u|| < ||\Pi_i - \Pi_u||$ where $||\ ||$ denotes the *variation distance* defined as the maximum of the difference of probabilities over all subsets, and Π_u denotes the uniform distribution on M. □

3 Analyzing the mixing rate of the genetic system

Let μ be the number of edges in G. We have to prove that $\mathcal{G}$ is "*NC rapid mixing*", which is equivalent to prove that $\mathcal{G}$ converges in at most a polylogarithmic number of generations. First we must define the initial distribution Π_0 for $\mathcal{G}$. Given the input graph G with μ edges, select uniformly and independently an edge e. Therefore $\mathcal{M}(\Pi_0) = \mathcal{M}$.

As we have already shown $\mathcal{M}(\Pi)$ is symmetric and ergodic for any distribution Π, therefore for any i, the transition matrix $\mathcal{A}(\Pi_i)$ of the Markov chain $\mathcal{M}(\Pi_i)$ has necessarily real eigenvalues. That means the matrix has a spectral representation in the form

$$\mathcal{A}(\Pi_i) = \sum_{k=0}^{N-1} \lambda_k(i) e^{(k)} e^{(k)^T}$$

where $\{e^{(k)}\}$ is an orthonormal basis of left eigenvectors of $\mathcal{A}(\Pi_i)$, with $\{\lambda_0, \lambda_1, \cdots, \lambda_{N-1}\}$ the set of eigenvectors ordered by decreasing value, where $\lambda_0 = 1$ and N is the number of matchings in G.

The matrix $E^{(k)} = e^{(k)} e^{(k)^T}$ is a *dyad*, i.e. it has rank 1, furthermore

$$E^{(i)} E^{(j)} = 0 \quad i \neq j$$
$$E^{(i)} E^{(i)} = E^{(i)}$$

Therefore we get the following property:

Lemma 2. *For any i, j we have*

$$\mathcal{A}(\Pi_i)\mathcal{A}(\Pi_j) = \sum_{k=0}^{N-1} \lambda_k(i)\lambda_k(j) E^{(k)}.$$

In the way we have defined the genetic system $\mathcal{G}$ we have the following relationship:

$$\begin{aligned}
\Pi_t =& \Pi_{t-1} A(\Pi_{t-1}) \\
=& \Pi_{t-2} A(\Pi_{t-2}) A(\Pi_{t-1}) \\
=& \Pi_{t-2} \sum_{k=0}^{N-1} \lambda_k(t-1)\lambda_k(t-2) E^{(k)}.
\end{aligned}$$

Now we can relate the eigenvalues of the different Markov chains that appear along the process with the eigenvalues of the initial chain.

Theorem 5. *For any $t \geq 0$ and $0 \leq k \leq N$, $\lambda_k(t) = \Theta\left([\lambda_k(0)]^{2^t}\right)$, where $\lambda_k(0)$ are the eigenvalues of $\mathcal{M}$*

Proof. The basis case is trivial, because $\mathcal{M}(\Pi_0) = \mathcal{M}$. Le us inductively assume that $\lambda_k(t') = \Theta\left([\lambda_k(0)]^{2^t}\right)$ for any $t' < t$, we get then

$$\begin{aligned}\Pi_t =& \Pi_{t-2} \sum_{k=0}^{N-1} \Theta\left([\lambda_k(0)]^{2^{t-2}}[\lambda_k(0)]^{2^{t-1}}\right) E^{(k)} \\ =& \Pi_{t-2} \sum_{k=0}^{N-1} \Theta\left([\lambda_k(0)]^{2^{t-1}+2^{t-2}}\right) E^{(k)}\end{aligned}$$

So, we get

$$\begin{aligned}\Pi_t =& \Pi_0 \sum_{k=0}^{N-1} \Theta\left([\lambda_k(0)]^{\sum_{1=1}^{t} 2^{t-i}}\right) E^{(k)} \\ =& \Pi_0 \sum_{k=0}^{N-1} \Theta\left([\lambda_k(0)]^{2^t}\right) E^{(k)}\end{aligned}$$

But we have a Markov process and therefore

$$A(\Pi_t) = \sum_{k=0}^{N-1} \Theta\left([\lambda_k(0)]^{2^t}\right) E^{(k)}$$

□

Now from Proposition 2.1 of [Sin93a] we known that

$$\frac{\lambda_{\max}(t)}{\min_{j \in M} \Pi_i(j)}$$

upper bounds the relative pointwise distance from the uniform distribution to the distribution to Π_t, therefore as this is the distribution at time t in the genetic system we get

Theorem 6. *The pointwise distance between the uniform distribution and the distribution of generation t in $\mathcal{G}$ at most*

$$\frac{\Theta\left(\lambda_{\max}^{2^t}(0)\right)}{\min_{j \in M} \Pi_i(j)}$$

Note that $\lambda_{\max}(0)$ is the second highest eigenvalue of a chain which is polynomially rapidly mixing, therefore the genetic system is exponentially rapidly mixing, and that implies logarithmic convergence

Theorem 7. *The genetic system $\mathcal{G}$ converges to the uniform distribution in $O(\log n)$ generations.*

4 NC simulation

In order to simulate the genetic system $\mathcal{G}$ in RNC, we need first to generate in RNC a matching according to Π_0, second to compute in RNC the mating operation and third to show that a sample of polynomial size is enough to carry on the simulation.

Sampling in RNC from the initial distribution can be done in constant time using as many processors as sample size.

Given two matchings m_l and m_r to compute in RNC the mating operation, that gives birth to child m_k, consider the following procedure:

Computation of a single step in $\mathcal{M}(\Pi)$

(1) Delete from both matchings, all edges in m_l that are also in m_r.

(2) Delete all edges in m_r that share a vertex with some edge in m_l.

(3) The matching m_k consists of the remaining edges in m_r and m_l.

To implement the above procedure with a PRAM, we represent a matching by two vectors of length n, where the i-th position is either 0, or the vertex matching vertex i, then the resources needed are a constant number of steps and linear number of processors.

Finally, to complete the proof that $\mathcal{G}$ can be simulated in RNC, we have to prove that we don't need an exponential size population to get a good estimate of the system $\mathcal{G}$, otherwise the number of processors needed would be exponential, notice that to generate in parallel a population of exponential size we would need to run in parallel an exponential number of copies of the above procedures. We may overcome this difficulty by showing that a polynomial size population suffices, thus in the overall scheme we will run a polynomial number of copies of processes that use a polynomial number of processors.

We define the following *restricted size* population model: Let s be a parameter to be determined later. We maintain at each time t, a population formed by a multiset F_t of s matchings. At $t = 0$ the initial population

is a random s-sample from Π_0. The population F_t at time t, we construct F_{t+1} by executing in parallel and independently s experiments, where each experiment consists of picking uniformly two parents from F_t, selecting who is the left parent with probability $1/2$, and apply the mating operator to both matchings to generate a new matching in F_{t+1}. Assume that the experiment finishes at time $t = f$, its value to be determined later.

Using a similar technique as the one described in [RSW92], the discrepancy between this model and the system $\mathcal{G}$ is captured by the concept of a *collision*. We say that a couple of nodes in the same generation *collide* if they share the same father. Given a matching m in F_f, we said that a node in previous populations is *active* if some its edges have been inherited by m. We define a *collision* in the derivation of m if there is a colliding couple such that their common parent is active.

Lemma 3. *The probability that the derivation of an element m in F_f has a collision is bounded by $2n^2 f/s$.*

Proof. Recall that we are interested in nodes having edges inherited in m. Hence, at each time t, there are at most n such nodes in F_t. If we fix a level t and consider all derivations such that no pair collides with an active node in levels 1 through t, then the candidates to have edges that end up in m, are selected at random uniformly and independently from $\{1, \ldots, s\}$. The chance of such colliding pair at level t is at most

$$\frac{1}{s}\binom{2n}{2} \leq 2n^2/s,$$

therefore for all t, given that there is no pair that collides with an active node at time less that t, the probability of having a pair that collides with an active node at time t, is at most $2n^2/s$. The lemma follows by summing over all $t \leq f$. □

Therefore, for any $\delta > 0$ the restricted size population system and $\mathcal{G}$ remain within variation distance δ for at least f steps provided that the population has size at least $\Theta\left(\frac{n^2 f}{\delta}\right)$. If we wish to have a variation distance smaller than $1/n^2$, a finite population of size $s = \Theta(n^4 \log n)$ suffices to do the simulation, with $f = \log n$. To finish the simulation, we remove any element from F_f that has a collision in its derivation. This procedure can be implemented in RNC. Thus the total number of parallel steps needed to implement the simulation is $O(\log n)$ with $O(n^7)$ processors, and we have proved the following result,

Theorem 8. *The system $\mathcal{G}$ can be simulated in RNC with probability greater than $1-\delta$, for any $\delta > 0$. Therefore, there exists a RNC uniform generator for the set of all matchings a graph.*

5 P-completeness

Consider the following problem given a graph G, a matching m, a finite sequence of edges $\sigma = (e_1, \ldots, e_k)$. Compute m^* the matching obtained in a walk from m using σ in $\mathcal{M}$.

We shall prove that the above problem is P-complete. To do so, we transform it into a decision problem, add an extra edge e, and rather than asking for m^* we ask whether $e \in m^*$. We call such problem *associated problem* for Markov chain $\mathcal{M}$.

Theorem 9. *The associated problem for chain $\mathcal{M}$ is P-complete.*

Proof. We present a reduction from the monotone alternating fan out two CVP. Given such a circuit $\alpha =< g_1, \ldots, g_r >$, we assume that gates are enumerated preserving the layered structure, that means inputs, followed by level 1 gates, followed by level 2 gates, ..., followed by the gates at the latest level. Assume that each gate is defined by the equation $g_k = g_i * g_j$ where $*$ represents AND or OR.

We will construct a bipartite graph, that has two vertices v_i, w_i associated to each gate i, and two additional nodes v_0 and w_0. We will assume that the graph is the complete bipartite graph, all v_i nodes are in one set and all w_i are in the other.

The initial matching m is as follows:

1-inputs and OR gates: each node is matched with it's twin, that means we have edges (v_i, w_i).

0-inputs and AND gates : each node is unmatched.

No more edges are added to m.

The sequence is constructed piecewise, a portion from each OR and AND gate, glued together in the gate ordering.

For an OR gate $g_k = g_i OR g_j$. We define the sequence in three blocks:

1. $(w_i, v_0), (w_0, v_j), (w_j, v_0), (w_0, v_i), (w_0, v_j), (w_i, v_0)$
2. $(v_k, w_k), (v_i, w_k), (v_k, w_j), (v_k, w_k)$
3. $(v_0, w_j), (w_0, v_i), (v_i, w_k), (w_j, w_k)$

For an AND gate $g_k = g_i AND g_j$ we add:

1. $(w_i, v_0), (w_0, v_j), (w_j, v_0), (w_0, v_i), (w_0, v_j), (w_i, v_0)$
2. $(v_i, w_j), (v_j, w_k), (w_i, v_k), (v_k, w_k)$
3. $(v_0, w_i), (w_0, v_i), (w_i, v_k), (v_j, w_k), (v_i, w_j)$

In figures 1 and 2 it is given the obtained matchings following, in three steps the three edge's blocks, in the four possible situations of input values, for OR and AND gates respectively.

It is easy to see that 0's propagate by non-connected nodes and 1's by connected ones. Therefore in the final matching we will have and edge joining the nodes associated to the last gate if and only if the circuit outputs true.

□

One must be careful to understand the last result. It says that unless $NC = P$, we can not simulate in NC a random walk on the described rapid mixing Markov Chain. It does not say anything about the NC simulation of a random walk.

6 Conclusions and Open Problems

We have given a RNC procedure to sample from an almost uniform distribution the set of all matchings in a graph. Our technique gives an *approximate* solution to an open problem in [MVV87] of sampling in parallel perfect matchings according to the uniform distribution. Approximate in the sense that our sampling is *almost uniform*, otherwise the polynomial hierarchy would collapse to P.

As we indicated the counting of matchings is a $\#P$-complete problem. Moreover the problem is self-reducible. Therefore an open problem is to devise a self-reducibility scheme for the problem, that together with the almost uniform generator presented in this paper, gives a Fully RNC Approximation Scheme for the counting of matchings.

References

[JaJ92] J. JaJa. *An introduction to parallel algorithms.* Addison-Wesley, 1992.

[JS89] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18:1149–1178, 1989.

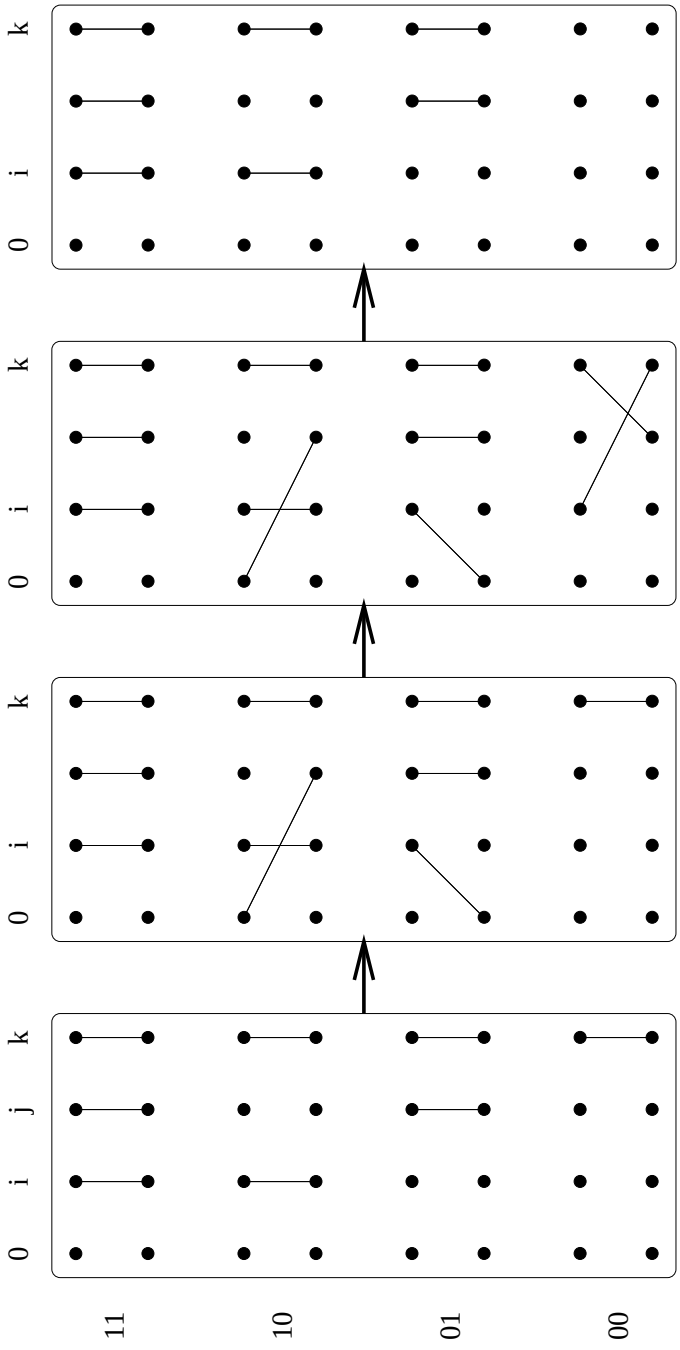

Figure 1: OR gates before and after processing its corresponding sequences.

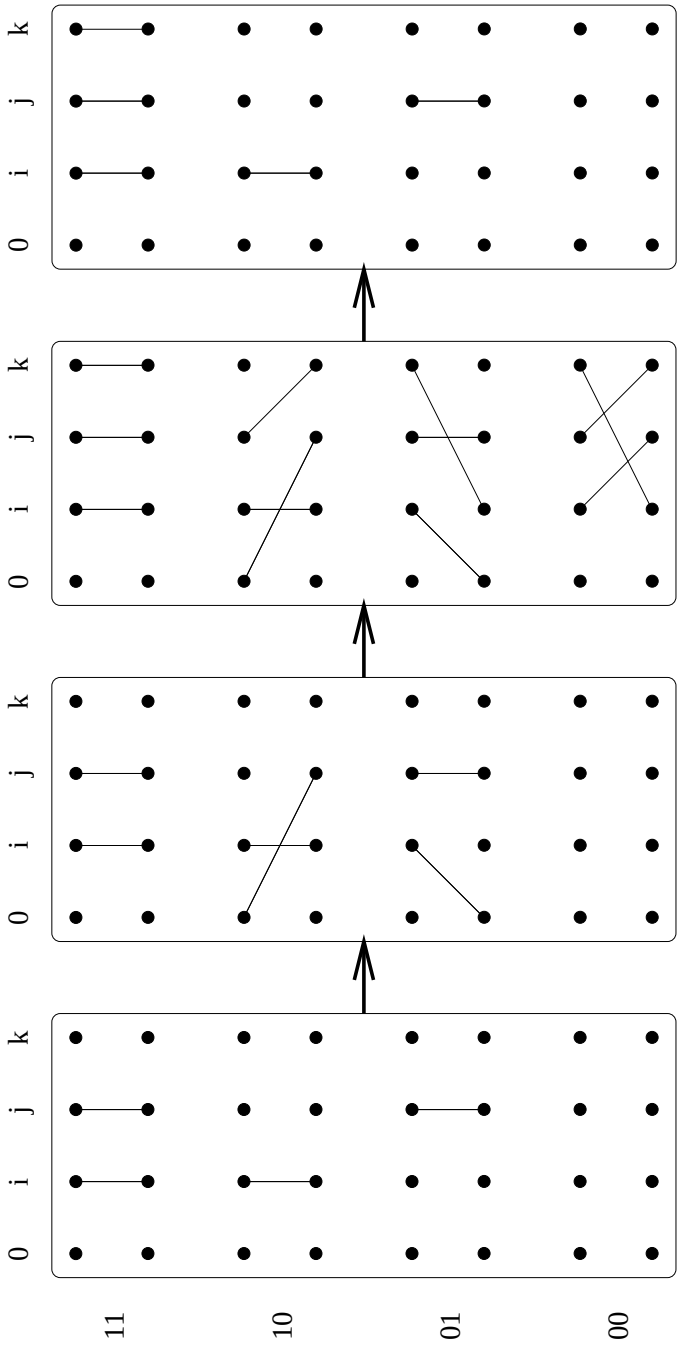

Figure 2: AND gates before and after processing its corresponding sequences.

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

[MR95] R. . Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, 1995.

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

[RRS95] Y. Rabani, Y. Rabinovich, and A. Sinclair. A computational view of population genetics. In *27th ACM Symposium on Theory of Computing*, pages 83–92, 1995.

[RSW92] Y. Rabinovich, A. Sinclair, and A. Wigderson. Quadratic dynamical systems. In *33th IEEE Symposium on Foundations of Computer Science*, pages 304–313, 1992.

[Ry70] A. Ren yi. *Probability Theory*. North-Holland, Amsterdam, 1970.

[Sin93a] A. Sinclair. *Algorithm for random generation and counting: A Markov chain approach*. Birkhäuser, Boston, 1993.

[Sin93b] A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommmodity flow. In *Combinatorics, Probability and Computing*, pages 351–370, Cambridge University Press, 1993.

[Ten95] Shang-Hua Teng. Independent sets versus perfect matchings. *Theoretical Computer Science*, pages 1–10, 1995.

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