18.409 An Algorithmist's Toolkit September 24, 2009

Lecture 5

Lecturer: Jonathan Kelner *Scribe: Shaunak Kishore*

1 Administrivia

Two additional resources on approximating the permanent

- Jerrum and Sinclair's original paper on the algorithm
- An excerpt from Motwani and Raghavan's *Randomized Algorithms*

2 Review of Monte Carlo Methods

We have some (usually exponentially large) set V of size Z, and we wish to know how many elements are contained in some subset S (which represents elements with some property we are interested in counting). A Monte Carlo method for approximating the size of S is to pick k elements uniformly at random from V and see how many are also contained in S. If q elements are contained in S, then return as our approximate solution Zq/k. In expectation, this is the correct answer, but how tightly the estimate is concentrated around the correct value depends on the size of $p = \frac{|S|}{|V|}$.

Definition 1 *An (ϵ, δ) **approximation scheme** is an algorithm for finding an approximation within a multiplicative factor of $1 \pm \epsilon$ with probability $1 - \delta$.*

Using the Chernoff bound, if we sample independently from a 0-1 random variable, we need to conduct

$$N \geq \Theta\left(\frac{\log \delta}{p\varepsilon^2}\right)$$

trials to achieve an (ϵ, δ) approximation, where $p = \frac{|S|}{|V|}$ as before. This bound motivates the definition of a polynomial-time approximation scheme.

Definition 2 *A **fully-polynomial randomized approximation scheme**, or **FPRAS**, is an (ϵ, δ) approximation scheme with a runtime that is polynomial in n, $1/\epsilon$, and $\log 1/\delta$.*

There are two main problems we might encounter when trying to design an FPRAS for a difficult problem. First, the S may be an exponentially small subset of V. In this case, it would take exponentially many samples from V to get $O(\frac{\log 1/\delta}{\epsilon^2})$ successes. Second, it could be difficult to sample uniformly from a large and complicated set V. We will see ways to solve both these problems in two examples today.

3 DNF Counting and an Exponentially Small Target

Suppose we have n boolean variables $x_1, x_2, \ldots x_n$. A literal is an x_i or its negation.

Definition 3 *A formula F is in **disjunctive normal form** if it is a disjunction (OR) of conjunctive (AND) clauses:*

$$F = C_1 \vee C_2 \vee \ldots \vee C_m,$$

where each C_i is a clause containing the ANDs of some of the literals. For example, the formula $F = (x_1 \wedge \overline{x_3}) \vee (x_2) \vee (x_2 \wedge \overline{x_1} \wedge x_3 \wedge x_4)$ is in disjunctive normal form.

If there are n boolean literals, then there are 2^n possible assignments. Of these 2^n assignments, we want to know how many of them satisfy a given DNF formula F. Unfortunately, computing the exact number of solutions to a given DNF formula is $\#P$-Hard[1]. Therefore we simply wish to give an ε-approximation for the number of solutions a given DNF formula that succeeds with probability $1 - \delta$ and runs in time polynomial in n, m, $\log \delta$, and $1/\varepsilon$.

Naïvely, one could simply try to use the Monte Carlo method outlined above to approximate the number of solutions. However the number of satisfying assignments might be exponentially small, requiring exponentially many samples to get a tight bound. For example the DNF formula $F = (x_1 \wedge x_2 \wedge \cdots \wedge x_n)$ has only 1 solution out of 2^n assignments.

3.1 Reducing the Sample Space

Instead of picking assignments uniformly at random and testing each clause, we will instead sample from the set of assignments that satisfy at least one clause (but not uniformly). This algorithm illustrates the general strategy of *sampling only the important space*.

Consider a table with assignments on one side and the clauses $C_1, C_2, \ldots, C_m$ on the other, where each entry is 0 or 1 depending on whether the assignment satisfies the clause. Then, for each assignment, we color the entry for first clause which it satisfies yellow (if such a clause exists). We color the remaining entries satisfied clauses blue, and we set these entries to 0. See Figure 1.

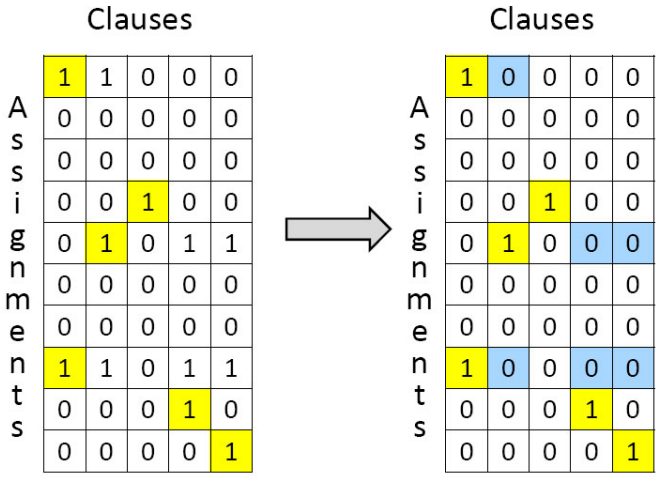


Figure 1: A table of assignments versus clauses and how to color it.

We will sample uniformly from the space of blue and yellow colored entries, and then test whether we've sampled a yellow entry. We then multiply the ratio we get by the total number of blue and yellow entries, which we can easily compute.

Let clause C_i have k_i literals. Then clearly the column corresponding to C_i has 2^{n-k_i} satisfying assignments (which we can easily compute). We choose which clause to sample from with probability proportional to 2^{n-k_i}. Then we pick a random satisfying assignment for this clause and test whether it is the first satisfied clause in its row (a yellow entry), or if there is a satisfied clause that precedes it (a blue entry). The total size of the space we're sampling from is just $\sum_i 2^{n-k_i}$.

[1]To see this, understand that the negation of a DNF formula is just a CNF formula by application of De Morgan's laws. Therefore counting solutions to a DNF formula is equivalent to counting (non-)solutions of a CNF formula, which is the canonical example of a $\#P$-Hard problem.

Our probability of picking a yellow entry is at least $1/m$, where m is the number of clauses, so we can take enough samples in polynomial time. Therefore, this algorithm is an FPRAS for counting the solutions to the DNF formula.

4 Approximating the Permanent of a 0-1 Matrix

Definition 4 (Determinant) *For a given $n \times n$ matrix M, the determinant is given by*

$$det(M) = \sum_{\pi \in S_n} sgn(\pi) \prod_{i=1}^{n} M_{i,\pi(i)}.$$

The formula for the permanent of a matrix is largely the same, with the $\text{sgn}(\pi)$ omitted.

Definition 5 (Permanent) *For a given $n \times n$ matrix M, the permanent is given by*

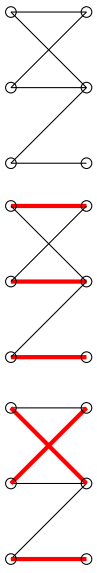
$$per(M) = \sum_{\pi \in S_n} \prod_{i=1}^{n} M_{i,\pi(i)}.$$

However, while the determinant of a matrix is easily computable — $O(n^3)$ by LU decomposition — calculating the permanent of a matrix is $\#P$-Complete. As we will show, computing the permanent of a 0-1 matrix reduces to the problem of finding the number of perfect matchings in a bipartite graph.

4.1 The Permanent of a 0-1 Matrix and Perfect Matchings

Given an $n \times n$ 0-1 matrix M, we construct a subgraph G of $K_{n,n}$, as follows. Let the vertices on the left be $v_1, v_2, \ldots v_n$ and let the vertices on the right be $w_1, w_2, \ldots w_n$. There is an edge between v_i and w_j if and only if M_{ij} is 1.

Suppose σ is a permutation of $\{1, 2, \ldots n\}$. Then the product $\prod_i M_{i\sigma(i)}$ is 1 if the pairing $(v_i, w_{\sigma(i)})$ is a perfect matching, and 0 otherwise. Therefore, the permanent of M equals the number of perfect matchings in G. As an example, we look at a particular 3×3 matrix and the corresponding subgraph of $K_{3,3}$.



$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \boxed{1} & 1 & 0 \\ 1 & \boxed{1} & 0 \\ 0 & 1 & \boxed{1} \end{pmatrix}$$

$$\begin{pmatrix} 1 & \boxed{1} & 0 \\ \boxed{1} & 1 & 0 \\ 0 & 1 & \boxed{1} \end{pmatrix}$$

Calculating the permanent of a 0-1 matrix is still $\#P$-Complete. As we will see, there is an FPRAS for approximating it.

4.2 An FPRAS for Approximating the Permanent of a Dense Graph

4.2.1 Some History

- 1989: Jerrum and Sinclair showed how to approximate the permanent of a dense graph (all vertices have degree at least $n/2$). At the time, it was not known if this result could be extended to the general case.
- 2001: Jerrum, Sinclair and Vigoda showed how to approximate the permanent of an arbitrary graph (and therefore for any matrix with nonnegative entries).

We will show today the result of 1989 for approximating the permanent of a dense graph.

4.2.2 General Strategy

We can't do the naïve Monte Carlo here, since the probability of picking a perfect matching from the set of all permutations can be exponentially small. Therefore we will instead consider the set of all (possibly partial) matchings, not just perfect ones. Let M_k be the set of all partial matchings of size k. Now suppose that we had a black box that samples uniformly at random from $M_k \cup M_{k-1}$ for any k. Then by the Monte Carlo method, by testing membership in M_k, we can determine the ratio $r_k = \frac{|M_k|}{|M_{k-1}|}$.

If we assume that for all k, $1/\alpha \leq r_k \leq \alpha$ for some polynomially-sized α, then we can estimate each r_k to within relative error $\varepsilon = 1/n^2$ using polynomially many samples. Therefore our estimate of the number of perfect matchings is just

$$|M_n| = |M_1| \prod_{i=2}^{n} r_i.$$

If all of our approximations were within a $(1 \pm \frac{1}{n^2})$ factor, then our total error is at most $(1 \pm \frac{1}{n^2})^n \approx (1 \pm \frac{1}{n})$.

4.2.3 Bounding the r_k

We first begin with a crucial lemma.

Lemma 6 *Let G be a bipartite graph of minimum degree $\geq n/2$. Then every partial matching in M_{k-1} has an augmenting path of length ≤ 3.*

Proof Let $m \in M_{k-1}$ be a partial matching. Let u be an unmatched node of m. Now suppose that there are no augmenting paths of length 1 starting from u in this matching m (i.e. there is no unmatched node v such that there is an edge connecting u and v). Then by our degree conditions, u must be connected to at least $n/2$ of the matched nodes v'_i. Likewise if we pick an unmatched node v, if it has no augmenting paths of length 1, then it must be connected to at least $n/2$ of the matched nodes u'_j. But by the pigeonhole principle, there must exist i and j such that $(u'_j, v'_i) \in m$. The path (u, v'_i, u'_j, v) is an augmenting path of length 3. ■

Theorem 7 *Let G be a bipartite graph of minimum degree $\geq n/2$. Then $1/n^2 \leq r_k \leq n^2$ for all k.*

Proof We first prove that $r_k \leq n^2$. Consider the function $f : M_k \to M_{k-1}$, which maps $m \in M_k$ to its (arbitrarily-chosen) canonical representative in M_{k-1} (i.e. uniquely choose a submatching of m). For any $m' \in M_{k-1}$, it must be the case that $|f^{-1}(m')| \leq (n-k+1)^2 \leq n^2$. Thus $|M_k| \leq n^2 |M_{k-1}|$.

Now we show that $1/n^2 \leq r_k$. Fix some $m \in M_k$. By Lemma 6, every partial matching in M_{k-1} has an augmenting path of length ≤ 3. There are at most k partial matchings in M_{k-1} that can by augmented by a path of length 1 to equal m. In addition, there are at most $k(k-1)$ matchings in M_{k-1} that can be augmented by a path of length 3 to equal m. Thus $|M_{k-1}| \leq (k + k(k-1))|M_k| = k^2 |M_k| \leq n^2 |M_k|$. ■

4.2.4 How to Sample (Approximately) Uniformly

We still have to show how to sample uniformly from $C_k = M_k \cup M_{k-1}$. We will only show how to sample *approximately* uniformly from this set. As it turns out, this result is good enough for our purposes.

The main idea here is to construct a graph whose vertex set is C_k, and then do a random walk on this graph which converges to the uniform distribution. We have to show two things: that the random walk converges in polynomial time, and that the stationary distribution on the graph C_k is uniform. To show that the random walk mixes quickly, we bound the conductance $\Phi(C_k)$ by the method of *canonical paths*.

Lemma 8 *Let $G = (V, E)$ be a graph for which we wish to bound $\Phi(G)$. For every $v, w \in V$, we specify a canonical path $p_{v,w}$ from v to w. Suppose that for some constant b and for all $e \in E$, we have*

$$\sum_{v,w \in V} \mathbf{I}[e \in p_{v,w}] \leq b|V|$$

that is, at most $b|V|$ of the canonical paths run through any given edge e. Then $\Phi(G) \geq \frac{1}{4bd_{max}}$, where d_{max} is the maximum degree of any vertex.

Proof As before, the conductance of G is defined as

$$\Phi(G) = \min_{S \subset V} \frac{e(S)}{\min\left\{\sum_{v \in S} d(v), \sum_{v \in \overline{S}} d(v)\right\}}.$$

Let $S \subset V$. We will show that $\Phi(S) \geq \frac{1}{4bd_{max}}$. Without loss of generality, assume that $|S| \leq |V|/2$. Then the number of canonical paths across the cut is at least $|S||\overline{S}| \geq |S||V|/2$. For each edge along the cut there can be no more than $b|V|$ paths through each edge, the number of edges $e(S)$ is at least $\frac{|S|}{2b}$.

In addition we can bound $\min\left\{\sum_{v \in S} d(v), \sum_{v \in \overline{S}} d(v)\right\}$ by $|S|d_{max}$. These bounds give us

$$\Phi(S) \geq \frac{|S|/2b}{|S|d_{max}} \geq \frac{1}{2bd_{max}}.$$

as claimed. ■

Since the spectral gap is at least $\Phi(G)^2$, as long as b and d_{max} are bounded by polynomials, a random walk on G will converge in polynomial time.

4.2.5 The Graph C_k

We will only do C_n. It should be clear later how to extend this construction for all k. Recall that our vertices correspond to matchings in $M_n \cup M_{n-1}$. We show how to connect our vertices with 4 different types of directed edges:

- **Reduce** ($M_n \longrightarrow M_{n-1}$): If $m \in M_n$, then for all $e \in m$ define a transition to $m' = m - e \in M_{n-1}$
- **Augment** ($M_{n-1} \longrightarrow M_n$): If $m \in M_{n-1}$, then for all u and v unmatched with $(u, v) \in E$, define a transition to $m' = m + (u, v) \in M_n$.
- **Rotate** ($M_{n-1} \longrightarrow M_{n-1}$): If $m \in M_{n-1}$, then for all $(u, w) \in m$, $(u, v) \in E$ with v unmatched, define a transition to $m' = m + (u, v) - (u, w) \in M_{n-1}$.
- **Self-Loop**: Add enough self-loops so that you remain where you are with probability 1/2 (this gives us a uniform stationary distribution).

Note that this actually provides an undirected graph since each of these steps is reversible.

Example 9 *In Figure 2 we show C_2 for the graph $G = K_{2,2}$. The two leftmost and two rightmost edges are **Augment/Reduce** pairs, while the others are **Rotate** transitions. The self-loops are omitted.*

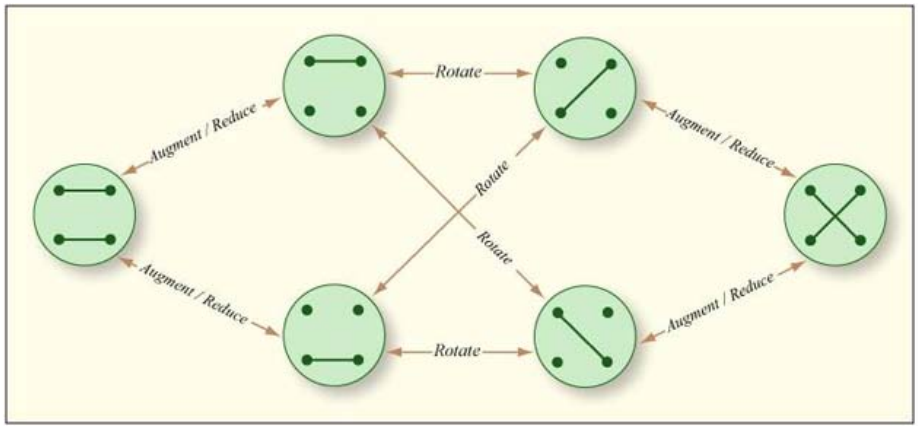


Figure by MIT OpenCourseWare.

Figure 2: C_2 for the graph $G = K_{2,2}$.

4.2.6 Canonical Paths

We still need to define the canonical paths $p_{v,w}$ for our graph C_n. For each node $s \in M_n \cup M_{n-1}$, we associate with it a "partner" $s' \in M_n$, as follows:

- If $s \in M_n$, $s' = s$.
- If $s \in M_{n-1}$ and has an augmenting path of length 1, augment to get s'.
- If $s \in M_{n-1}$ and has a shortest augmenting path of length 3, augment to get s'.

Now for nodes $s, t \in M_n \cup M_{n-1}$, we show how to provide a canonical path $p_{s,t}$ which consists of three segments (and each segment can be one of two different types).

- $s \longrightarrow s'$ (Type A)
- $s' \longrightarrow t'$ (Type B)
- $t' \longrightarrow t$ (Type A)

Type A paths are paths that connect a vertex $s \in M_n \cup M_{n-1}$ to its partner $s' \in M_n$. Clearly, if $s \in M_n$ then the type A path is empty. Now if $s \in M_{n-1}$ and has an augmenting path of length 1, then our canonical path is simply the edge that performs the **Augment** operation. If $s \in M_{n-1}$ and has a shortest augmenting path of length 3, then our canonical path is of length 2: first a **Rotate**, then an **Augment** (see Figure 3 for an example).

For a Type B path, both s' and t' are in M_n. We let $d = s' \oplus t'$, the *symmetric difference* of the two matchings (those edges which are not common to both matchings). It is clear that since s' and t' are perfect matchings, d consists of a collection of disjoint, even-length, alternating (from s' or from t') cycles of length at least 4.

Our canonical path from s' to t' will in a sense "unwind" each cycle of d individually. Now, in order for the path to be canonical, we need to provide some ordering on the cycles so that we process them in the same order each time. However, this can be done easily enough. In addition, we need to provide some ordering on the vertices in each cycle so that we unwind each cycle in the same order each time. Again, this can be done easily enough. All that remains is to describe how the cycles are unwound, which can be done much more effectively with a picture than by text. See Figures 4 and 5.

We must now bound the number of canonical paths that pass through each edge. First we consider the type A paths.

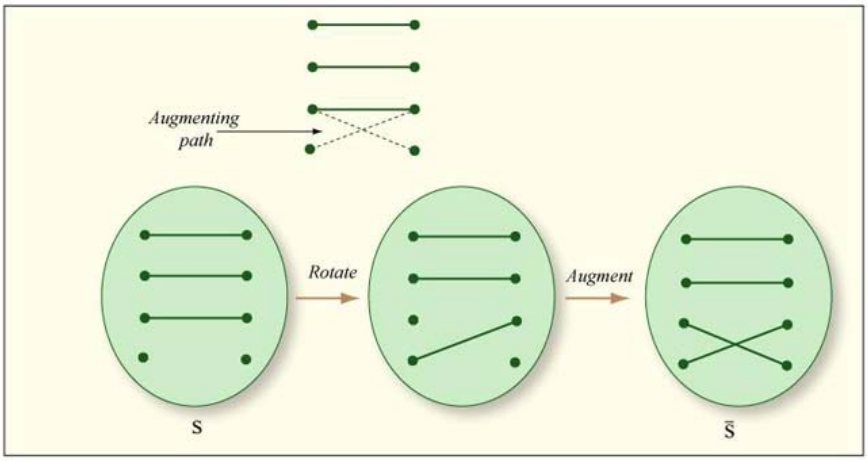


Figure by MIT OpenCourseWare.

Figure 3: Type A path of length 2.

Lemma 10 *Let $s \in M_n$. Then at most $O(n^2)$ other nodes $s' \in M_n \cup M_{n-1}$ have s as their partner.*

Proof There are three possible types of nodes s' that have s as their partner. The first is if $s' = s$ (hence the type A path is empty). The second can be obtained by a **Reduce** transition (the nodes s' with augmenting path of length 1). The third can be obtained by a **Reduce** and **Rotate** pair of transitions. There is only partner for the first, $O(n)$ for the second, and $O(n^2)$ for the third. Therefore there are at most $O(n^2)$ nodes s' that can count s' as their partner. ■

Now we wish to count the number of canonical paths for type B.

Lemma 11 *Let T be a transition (i.e. an edge of C_n). Then the number of pairs $s, t \in M_n$ that contain T on their type B canonical path is bounded by $|C_n|$.*

Proof We will provide an injection $\sigma_T(s,t)$ that maps to matchings in $C_n = M_n \cup M_{n-1}$. As before, let $d = s \oplus t$ be the symmetric difference of the two matchings s and t (recall that these can be broken down into disjoint alternating cycles $C_1, \ldots, C_r$). Now we proceed along the unwinding of these cycles until we reach the transition T. At this point we stop and say that the particular matching we are at, where all cycles up to this point agree with s and all cycles after this point agree with t, is the matching that $\sigma_T(s,t)$ maps to.

It is clear that this is fine when T is a **Reduce** or **Augment** transition, since these only occur at the beginning or end of an unwinding. The only problem is when T is a **Rotate** transition, because then there exists a vertex u (the pivot of the rotation) that is matched to a vertex v with $(u,v) \in s$ and is also matched to a vertex w with $(u,w) \in t$. This is because up to T we agree with s, and after T we agree with t. But what we can do at this point is notice that one of these two edges (which we denote by $e_{s,t}$) always has the start vertex of the current cycle as one of its end-points. Therefore by removing it we end up with a matching again. This is further illustrated in Figure 6. ■

Theorem 12 *The conductance of our graph has the following bound*

$$\Phi(C_n) = \Theta\left(\frac{1}{n^6}\right)$$

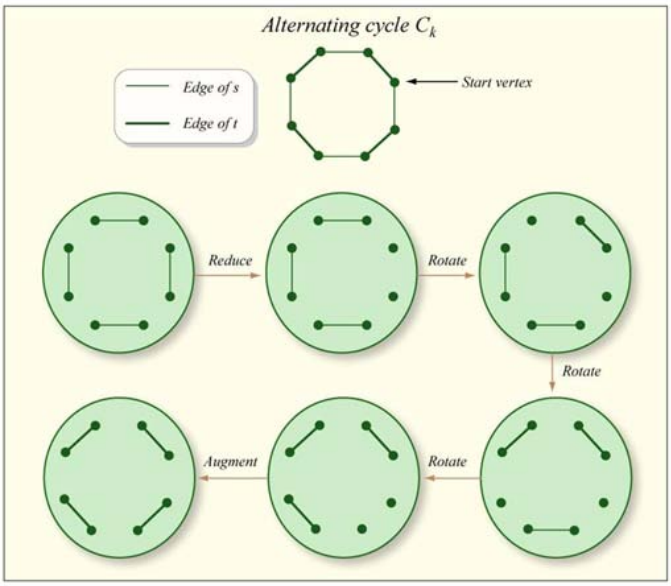


Figure 4: Unwinding a single cycle (type B path).

Proof By Lemma 8, we have $\Phi(G) \geq \frac{1}{2bd_{max}}$. As shown in Lemma 10, there are at most $O(n^2)$ canonical paths to $s \in M_n$ from M_{n-1}, and at most $O(n^2)$ canonical paths from $t \in M_n$ to M_{n-1}. In addition we showed in Lemma 11 that the number of type B paths through a particular transition T is bounded by $|M_n \cup M_{n-1}| = |V|$ (where V is the vertex set of C_n). Therefore as a whole, the number of canonical paths through a particular transition T is bounded by $n^2 \times |V| \times n^2$, which implies $b = n^4$.

Since $d_{max} = O(n^2)$, the conductance is bounded from below by $\Theta\left(\frac{1}{n^6}\right)$ and our random walk mixes in polynomial time. ■

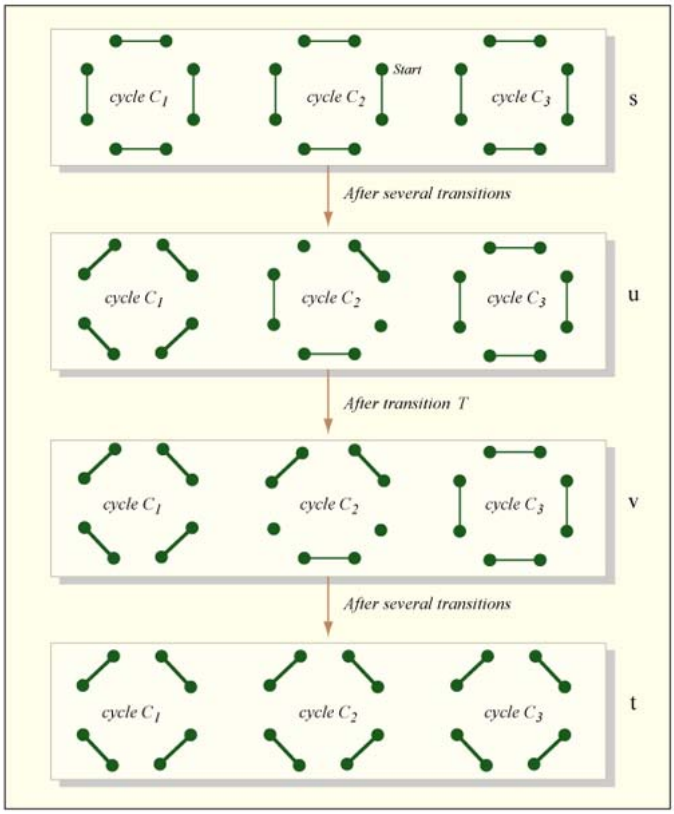


Figure 5: Unwinding a collection of cycles (type B path).

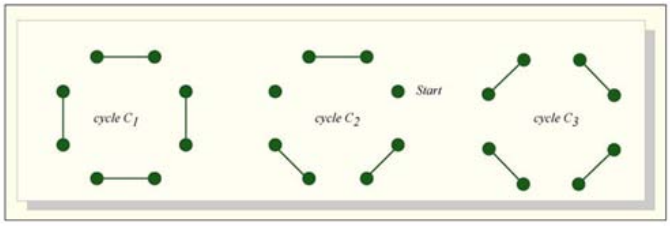


Figure 6: The encoding $\sigma_T(s,t)$.

18.409 Topics in Theoretical Computer Science: An Algorithmist's Toolkit
Fall 2009