Parallel and Distributed Algorithms

Winter 2009/10

Prof. Dr. Georg Schnitger

JOHANN WOLFGANG GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Arbeitsgruppe Theoretische Informatik, Institut für Informatik

---

Assignment 6

Issue: 23.11.2009
Due: 30.11.2009

6.1. Problem (6) *Monte Carlo Approximation of π*

We show how to approximate π. Observe that the area of the circle C with radius one equals π, whereas the area of the square $S = [-1, +1]^2$ equals four. Hence if we randomly draw p points from S and if $p(C)$ is the number of points which already belong to the circle C, then the ratio $\frac{p(C)}{p}$ converges to $\frac{\pi}{4}$.

Assume that we draw n points from the square S at random.

a.) Use the Chebycheff inequality to bound the error $\frac{|\text{estimate} - \pi/4|}{\pi/4}$.

b.) Apply the Chernoff inequality to bound the same error.

6.2. Problem (6) *Random Walk of limited length*

Let $\mathcal{M} = (\Omega, P)$ be a Markov chain. Show that $P^t[u, v]$ is the probability that the random walk reaches state v after t steps provided the walk begins in state u.

6.3. Problem (6) *Stationary distribution*

Determine the stationary distribution of the Markov chain with transition matrix P. Is the stationary distribution unique?

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{3}{4} \end{pmatrix}$$

6.4. Problem (14) *Random Walk vs 2Sat*

(a) We consider a Markov chain with state set $\{0, 1, \ldots, n\}$ and transition probabilities $P[i, i+1] = P[i, i-1] = 1/2$ for $1 \leq i \leq n-1$ and $P[0, 1] = P[n, n] = 1$. Let $T(i)$ be the expected time to reach state n when beginning in state i. Show that $T(i)$ equals $n^2 - i^2$.

(b) A formula $\alpha = c_1 \wedge c_2 \wedge \ldots \wedge c_m$ is a 2CNF formula (i.e. in conjunctive normalform with two literals per clause), if each clause c_i is of the form $c_i = (l_{i,1} \vee l_{i,2})$, where $l_{i,1}$ and $l_{i,2}$ are variables or their negations.

We have to determine whether a 2CNF formula α is satisfiable: α is satisfiable iff there is an assignment of truth values to the variables such that each clause is satisfied.

To determine satisfiability of a given 2CNF formula, our algorithm first chooses an arbitrary assignment x of truth values. Then it repeats the following steps *sufficiently often*: if the current assignment is satisfying, the algorithm halts. Otherwise determine an arbitrary clause which is not satisfied and flip the value of a randomly selected variable appearing in the clause. If no satisfying assignment is found, declare α to be not satisfiable.

Use part (a) to determine the expected number of iterations sufficient to find a satisfying assignment with probability at least $1/2$ bounded by $O(n^2)$, provided α is satisfiable.