Jinyang Ruan

011696096

CPT_S 515 Advanced Algorithms HW#6

1. After testing this coffee maker, we have finite IO sequences.

   Drop off all outputs and keep all inputs.

   Let S be a finite set of sequences of input symbols.

   Build a de Bruijn graph out of all the sequences in S.

   Then, normalize the graph into a Markov chain.

   For each sequence α in S, walk the sequence along the Markov chain. Suppose the length of α is k. We will get the transition probabilities $p_1, p_2, \dots, p_k$.

   Define $\lambda_\alpha = -\log(p_1 \cdot p_2 \cdots p_k)\ and\ \lambda^* = \frac{1}{|S|}\sum_{\alpha \in S} \lambda_\alpha$.

   Define ε is a small positive number. Then, if one sequence α satisfies $|\lambda_\alpha - \lambda^*| < \varepsilon$, α is considered as “random”.

   I put the answer I got from your paper in the last page, I wrote it before you give us hint. I am still trying to understand it. λ^* in paper seems to be the entropy of the most “random” walk in graph rather than the average of all λ_α.

3. In order to proof #3SAT is #P-complete. I studied some definitions and theorems from Berkeley public notes [2].

   Definition 1: *R is an NP-relation if there is a polynomial time algorithm A such that* $(x, y) \in R \Leftrightarrow A(x, y) = 1$ *and there is a polynomial p such that* $(x, y) \in R \Rightarrow |y| \le p(|x|)$.

$\#R$ is the problem that, given x, asks how many y satisfy $(x, y) \in R$.

Definition 2: *#P is the class of all problems of the form #R, where R is an NP-relation.*

Definition 3: *We say there is a parsimonious reduction from #A to #B (written $\#A \leq_{par} \#B$) if there is a polynomial time transformation f such that for all x,* $|\{y, (x, y) \in A\}| = |\{z : (f(x), z) \in B\}|$.

Often this definition is a little too restrictive and we use the following definition instead.

Definition 4: $\#A \leq \#B$ *if there is a polynomial time algorithm for* $\#A$ *given an oracle that solves* $\#B$.

Then I studied #CIRCUITSAT. #CIRCUITSAT is the problem where given a circuit, we want to count the number of inputs that make the circuit output 1.

Theorem: *#CIRCUITSAT is #P-complete under parsimonious reductions*.

Proof. Let #R be in #P and A and p be as in the definition. Given x we want to construct a circuit C such that $|\{z : C(z)\}| = |\{y : |y| \leq p(|x|), A(x, y) = 1\}|$. We then construct $\hat{C}_n$that on input x, y simulates $A(x, y)$. From earlier arguments we know that this can be done with a circuit with size about the square of the running time of A. Thus, $\hat{C}_n$ will have size polynomial in the running time of A and so polynomial in x. Then let $C(y) = \hat{C}(x, y)$.

Finally, there is a theorem that shows *#3SAT is #P-complete.*

Proof. We show that there is a parsimonious reduction from #CIRCUITSAT to #3SAT. That is, given a circuit C, we construct a Boolean formula φ such that the number of satisfying assignments for φ is equal to the number of inputs for which C

outputs 1. Suppose C has inputs $x_1, \dots, x_n$ and gates $1, \dots, m$ and φ has inputs $x_1, \dots, x_n$, $g_1, \dots, g_m$, where the g_i represent the output of gate i. Now each gate has two input variables and one output variable. Thus, a gate can be complete described by mimicking the output for each of the 4 possible inputs. Thus, each gate can be simulated using at most 4 clauses. In this way we have reduced C to a formula φ with $n + m$ variables and $4m$ clauses. So, there is a parsimonious reduction from #CIRCUITSAT to #3SAT.

4. 3SAT is NP-complete problem.

There are n variables $x_1, \dots, x_n$. k is the size of a cover C of a given F. Say $C = \{y_1, \dots, y_k\}$

For each assignment $b_1, \dots, b_k$ to $y_1, \dots, y_k$, // time $= O(2^k)$

Check $F|_{y_1=b_1,\dots,y_k=b_k}$ is satisfied or not. // 2SAT, time $= O(p(n))$

if yes, return YES

otherwise, return NO.

One way to solve 2SAT problem:

Build a directed graph $G =< V, E >$ with $2 \cdot n$ nodes, each pair of vertices v_i, v_{i+1} corresponds to the values 0 and 1 of $x_i \in S$.

If $x_i \vee x_j = 1$, construct edges $e_{i,j+1}, e_{j,i+1}$

If $x_i \vee x_j = 0$, construct edges $e_{i+1,j}, e_{j+1,i}$

If $x_i \bigoplus x_j = 1$, construct edges $e_{i,j+1}, e_{j,i+1}, e_{i+1,j}, e_{j+1,i}$

A specific directed graph $G \; =< \; V, E \; >$ can be constructed by specific constraints. When a vertex v is selected in the graph, other corresponding vertices must be selected in order to meet the constraint conditions. All the strongly connected branches in the digraph constructed by 2SAT are all the solutions of the 2SAT problem. The problem can be transformed into the solution of the strongly connected branches of the digraph. Using Tarjan’s algorithm, and it takes polynomial time.

Thus, solving the problem of 3SAT takes $O(2^k \cdot p(n))$ which is exponential in k and polynomial in n. We say that problem is fixed parameter tractable wrt k.

References

[1] Cui, Cewei, Dang, Zhe, & Fischer, Thomas R. (2011). Typical Paths of a Graph. *Fundamenta Informaticae*, 110(1-4), 95–109. https://doi.org/10.3233/FI-2011-530

[2] Trevisan, Luca. (2012). Retrieved from: https://people.eecs.berkeley.edu/~luca/cs254-12/lecture02.pdf.

1. After testing this coffee maker, we have finite IO sequences.

We define a finite graph G has a set of nodes Q and a set of directed edges E. Without loss of generality, we assume that every node is reachable from the initial node q_1. One IO sequence can be considered as a finite sequence of nodes in G [1].

In this question, each output is produced by an input preceding it, so we can use Markov chain to represent the process of IO sequence. A finite state Markov chain $\mathcal{X}$ is a discrete stochastic process $X_1, X_2, \dots X_n, \dots$ where the sample space for each random variable X_n is Q, and the conditional probability of $\mathcal{X}$ needs to satisfy:

$$Prob(X_n = x_n | X_{n-1} = x_{n-1}, \dots, X_1 = x_1) = Prob(X_n = x_n | X_{n-1} = x_{n-1})$$

for all $x_1, x_2, \dots, x_n \in Q$. Together with the initial distribution $Prob(X_1 = x_1) = 1$, the probability of a particular sequence $\pi = x_1, x_2, \dots, x_n$ for some $n \geq 1$ is:

$$Prob(\pi) = Prob(X_1 = x_1) \cdot Prob(X_2 = x_2 | X_1 = x_1) \cdots Prob(X_n = x_n | X_{n-1} = x_{n-1})$$

Hence, the Markov chain can also be represented in the form of *probability transition matrix* $T = [T_{ij}]$ where each T_{ij} indicates the *transition probability* $Prob(X_{n+1} = q_j | X_n = q_i)$ from the node q_i to q_j. The Markov chain $\mathcal{X}$ is called a *G-represented* Markov chain if, for each T_{ij}, $T_{ij} = 0$ where there is no edge from node q_i to q_j in *G*.

In information theory, entropy rate indicates the growth rate of the entropy of a stochastic process. The entropy rate of $\mathcal{X}$ is defined as:

$$\lambda_{\mathcal{X}} = \limsup_{n \to \infty} \frac{1}{n} H(X_1, \dots, X_n)$$

where $H(X_1, \dots, X_n)$ is the joint entropy of $X_1, \dots, X_n$, defined as, according to Shannon,

$$\sum_{x_1,\ldots,x_n} p(x_1, \ldots, x_n)\, log \frac{1}{p(x_1, \ldots, x_n)}$$

Notice that the upper limit always exists when the Markov chain $\mathcal{X}$ is *G-represented*. In this case, the maximal rate λ^* among all possible *G-represented* $\mathcal{X}$ is shown that:

$$\lim_{n\to\infty} \frac{\log(S_n)}{n} = \lambda^*$$

where S_n is the number of paths in G with length n, and the rate λ^* is achievable by a *G-represented* $\mathcal{X}$. λ^* refers to the entropy rate of a most “random” Markov walk on the graph.

Let a positive small number $\varepsilon > 0$. For a chosen test sequence and its Markov walk $\mathcal{X}$, if

$$|\lambda_{\mathcal{X}} - \lambda^*| < \varepsilon$$

which means this test sequence is “similar” to the most “random” Markov walk on the graph. We call this test sequence is random.