| | |
|---|---|
| **B62005Y-02 理论计算机科学基础** | 2017 年 7 月 3 日 |

# Solutions to Homework #8

| | |
|---|---|
| 主讲教师：杨光 | 张远航 2015K8009929045 |

**Sipser 9.8** $R^{\{m,n\}} = R \uparrow m \circ ((R \cup \varepsilon) \uparrow (n-m))$.

**Sipser 9.11** *Proof.* Since *CLIQUE* is NP-complete, *CLIQUE* is polynomial time reducible to *SAT*. For any instance $\langle G, k\rangle$ of *MAX-CLIQUE* we simply use the reduction on $\langle G, k\rangle$ and $\langle G, k+1\rangle$ and query the oracle for *SAT*. If exactly the former accepts while the latter rejects, *accept*; otherwise, *reject*. $\square$

**Sipser 9.14** We use the linear-size serial adder described in 9.12. Let $n = 2^k$. Then we have an recursive equation for the size of the entire circuit:

$$T(n) = T\left(\frac{n}{2}\right) + k$$

Solving for $T(n)$ we have $T(n) = O(2^k)$, which suggests that the circuit is of $O(n)$ size.

**Sipser 9.24** *Proof.* Suppose to the contrary that $TQBF \in \mathrm{SPACE}(n^{1/3})$. By the space hierarchy theorem, there is language $L \in \mathrm{SPACE}(n^{1+\varepsilon}) \setminus \mathrm{SPACE}(n)$. Using the reduction in Theorem 8.9 which shows that *TQBF* is NP-complete, any input to $L$ can be reduced to one of length $O(n^{2(1+\varepsilon)})$. Then $L \in \mathrm{SPACE}(n^{\frac{2}{3}(1+\varepsilon)})$. However, if we pick $\varepsilon$ such that $0 < \varepsilon < \frac{1}{2}$, we would have $L \in \mathrm{SPACE}(n)$, contradicting our choice of $L$. $\square$

**Sipser 10.2** $2^{12-1} \not\equiv 1 \pmod{12}$.

**Sipser 10.8** *Proof.* For any language $L \in \mathrm{BPL}$ there is a probabilistic log-space TM $M$ that decides $L$ with error probability greater than $\frac{1}{3}$. On input $w$ of length, let $C$ be the number of configurations of $M$ running on $x$, starting from any state. Construct a $C \times C$ matrix $P$ such that $P_{c_1,c_2} = \frac{1}{2}$ if $c_2$ is reachable from $c_1$ in one step and $P_{c_1,c_2} = 0$ otherwise. For every $t$, $P^t_{c_1,c_2}$ is the probability of reaching configuration $c_2$ from configuration $c_1$ in $t$ steps. By computing all powers of $P$ up to the running time of $M$ we can compute the accepting probability of $M$ starting from $q_0$, and decide if $x \in L$. Finally, we accept if $P^t_{c_0,c_{\mathrm{accept}}} > \frac{2}{3}$. $\square$

**Sipser 10.11** *Proof.* Assume NP $\subseteq$ BPP. Note that RP $\subseteq$ NP unconditionally, hence we just need to show that NP $\subseteq$ RP. Since $SAT$ is NP-complete and RP is closed under polynomial-time mapping reductions, it is enough to show that $SAT$ is in RP.

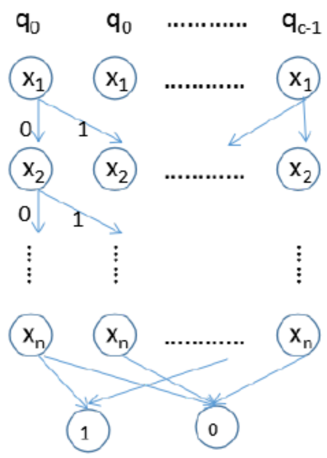
By assumption $SAT$ is in BPP. Let $M$ be a probabilistic Turing machine running in polynomial time and accepting $SAT$ with error at most $1/3$. Using the amplification lemma, we can define a probabilistic Turing machine $M'$ running in polynomial time and accepting SAT with error at most $2^{-\Omega(n)}$, where $n$ is the input length. We will define a probabilistic Turing machine $N$ running in polynomial time, which on input $\phi$, accepts with probability at least $1/2$ if $\phi$ is satisfiable, and accepts with probability 0 if $\phi$ is unsatisfiable.

The basic idea is to use *downward self-reducibility* to construct a satisfying assignment with high probability for YES instances, and to accept only if a satisfying assignment has been constructed. More precisely, $N$ does the following on an input $\phi$. Assume without loss of generality that the variables in $\phi$ are $x_1, x_2 \ldots x_m$.$N$ first runs $M$ on $\phi$. If $M$ rejects, $N$ rejects. Otherwise $N$ sets $x_1$ to FALSE in $\phi$ and runs $M$ on the corresponding formula $\phi_0$. If $M$ accepts, $N$ continues to build a satisfying assignment by setting $x_2$ to FALSE in $\phi_0$ and running $M$ on the corresponding formula $\phi_{00}$. If $M$ rejects, $N$ runs $M$ on formula $\phi_1$ obtained by setting $x_1$ to TRUE in $\phi$, continuing to build an assignment if $M$ accepts on $\phi_1$ and rejecting otherwise. This process continues until either $N$ rejects or all variables are set. If the latter, $N$ checks whether the corresponding assignment satisfies $\phi$. If yes, it accepts, otherwise it rejects.

$N$ runs in polynomial time since it makes at most a linear number of calls to the polynomial-time probabilistic TM $M$, and each of these calls is on an input whose length is at most the length of $\phi$ (setting variables can only decrease the length of a formula). $N$ never accepts on an unsatisfiable formula, hence all we need to show is that $N$ accepts with probability at least $1/2$ on a satisfiable formula. Note that if $M$ always gives correct answers on calls to $M$, then when $\phi$ is satisfiable, $N$ constructs a satisfying assignment to $\phi$ and hence accepts. The probability that this happens is at least $1 - n \cdot 2^{-\Omega(n)}$, which is at most $1/2$ for large enough $n$, since the probability that $M$ gives at least one wrong answer is at most $n \cdot 2^{-\Omega(n)}$ by the union bound. $\square$

**Sipser 10.12** We construct the branching program as follows. Since $A$ is regular, there is a DFA that recognizes $A$. Let $n$ be the numbers of states. For each node $j$ of level

$i = 1, 2, \cdots, n-1$, add transitions to node $s$ and node $t$ of layer $i+1$, corresponding to $\delta(q_j, 0)$ and $\delta(q_j, 1)$. For the last layer, we point each node $j$ to 1 if $\delta(q_j, 0)$ or $\delta(q_j, 1)$ is an accept state and 0 otherwise.




（盗用一下助教姐姐的图……）

**Sipser 10.23** By Exercise 10.12 we know that $A$ has a size $O(n)$, depth $O(n)$ circuit. We can simply apply the circuit recursively.