Problem Set 6 Solutions

In all problems the languages are over the alphabet $\Sigma = \{0, 1\}$. If $x, y$ are strings then $x \parallel y$ denotes their concatenation. If $i \geq 0$ is an integer then $0^i$ denotes the string of $i$ zeros and $1^i$ denotes the string of $i$ ones. (If $i = 0$, both equal $\varepsilon$). If $x$ is a string then $|x|$ denotes its length. The Notes on Randomized Algorithms and Decision versus Search, available from the course webpage, are relevant to this problem set.

Problem 1. [40 points] We associate to languages $A, B$ the language

$$J(A, B) = \{ a \parallel b : a \parallel 0^{|b|} \in A \text{ and } 1^{|a|} \parallel b \in B \}.$$ 

Prove that if $A, B \in \text{BPP}$ then $J(A, B) \in \text{BPP}$.

Since $A \in \text{BPP}$, there is a polynomial-time, randomized parameterized TM $M_A$ recognizing $A$ with two-sided error. Also since $B \in \text{BPP}$, there is a polynomial-time, randomized parameterized TM $M_B$ recognizing $B$ with two-sided error. Let $r_A$ be the number of coins used by $M_A$ and $r_B$ the number used by $M_B$. We define the following randomized TM $M$:

$$M(x; R)$$

$$n \leftarrow |x| ; \ k \leftarrow \max( 3, \ \lceil \log_2[4(n+1)] \rceil )$$

For $i = 0, \ldots, n$ do

Let $R_A^i$ be the next $r_A(|\langle x, 1^k \rangle|)$ bits of $R$

Let $R_B^i$ be the next $r_B(|\langle x, 1^k \rangle|)$ bits of $R$

If $M_A(\langle x[1..i]\parallel 0^{n-i}, 1^k \rangle; R_A^i)$ accepts and $M_B(\langle 1^i \parallel x[i+1..n], 1^k \rangle; R_B^i)$ accepts then accept

EndFor

Reject

We are denoting by $x[l..m]$ the string consisting of bits $l$ through $m$ of string $x$, with the convention that when $m < l$ this is the empty string. We are denoting by $\log_2(\cdot)$ the logarithm in base 2. The machine above invokes each of $M_A, M_B$ $n+1$ times and hence runs in polynomial time. We now show that it recognizes $J(A, B)$ with two-sided error-probability $1/4$. This shows that $J(A, B) \in \text{BPP}$.

Suppose $x \in J(A, B)$. This means there is an $i \in \{0, \ldots, n\}$ such that $x[1..i]\parallel 0^{n-i} \in A$ and $1^i \parallel x[i+1..n] \in B$. But $\text{AccPr}_M(x)$ is at least the probability that the $i$-th iteration of the For loop of $M$ accepts, meaning

$$\text{AccPr}_M(x) \geq \text{AccPr}_{M_A}(\langle x[1..i]\parallel 0^{n-i}, 1^k \rangle) \cdot \text{AccPr}_{M_B}(\langle 1^i \parallel x[i+1..n], 1^k \rangle)$$
\[
\begin{align*}
\geq & \ (1 - 2^{-k})^2 \\
= & \ 1 - 2 \cdot 2^{-k} + 2^{-2k} \\
\geq & \ 1 - 2^{-k} \\
= & \ 1 - 2^{-(k-1)}.
\end{align*}
\]

Now suppose \(x \notin J(A,B)\). This means that for each \(i \in \{0, \ldots, n\}\), either \(x[1..i]|0^{n-i} \not\in A\) or \(1^i|x[i+1..n] \not\in B\). For each \(i\) define the event
\[
E_i : M_A(\langle x[1..i]|0^{n-i}, 1^k \rangle; R^i) \text{ accepts and } M_B(\langle 1^i|x[i+1..n], 1^k \rangle; R^i) \text{ accepts}.
\]

Then
\[
\text{AccPr}_M(x) = \Pr [E_0 \lor E_1 \lor \ldots \lor E_n] \tag{1}
\]
\[
\leq \sum_{i=0}^{n} \Pr [E_i] \tag{2}
\]
\[
= \sum_{i=0}^{n} \text{AccPr}_{M_A}(\langle x[1..i]|0^{n-i}, 1^k \rangle) \cdot \text{AccPr}_{M_B}(\langle 1^i|x[i+1..n], 1^k \rangle) \tag{3}
\]
\[
\leq \sum_{i=0}^{n} 2^{-k} \tag{4}
\]
\[
\leq (n + 1) \cdot 2^{-k}.
\]

Let us justify the numbered steps above. Equation (1) is true because \(M\) accepts exactly if either event \(E_0\) happens, or event \(E_1\) happens, or \ldots or event \(E_n\) happens. Equation (2) is a standard and basic fact of probability theory, namely that the probability of the OR of events is bounded above by the sum of the probabilities of these events.\(^1\) Equation (3) is true by the definition of \(E_i\) and the code of \(M\). Equation (4) is true because \(M_A\) is a parameterized TM recognizing \(A\) with two-sided error-probability and \(M_B\) is a parameterized TM recognizing \(B\) with two-sided error-probability.

For correctness it thus suffices that \(1 - 2^{-(k-1)} \geq 3/4\) and \((n + 1) \cdot 2^{-k} \leq 1/4\). We check that our choice of \(k\) guarantees this.

Since \(k \geq 3\), we have \(2^{-(k-1)} \leq 2^{-(3-1)} = 1/4\) and thus \(1 - 2^{-(k-1)} \geq 1 - 1/4 = 3/4\), showing the first constraint is met.

Since \(k \geq \log_2[4(n+1)]\) we have \(2^{-k} \leq 1/[4(n+1)]\), and thus \((n+1)2^{-k} \leq 1/4\), showing the second constraint is met.

---

Problem 2. [50 points] Prove that

1. [40 points] If SAT \(\in\ BPP\) then SAT \(\in\ RP\).

The assumption that SAT is in BPP together with the error-reduction results we have seen about BPP tell us that there is a polynomial-time, randomized parameterized algorithm \(B\)

---

\(^1\) Think of the simple case of two events \(A, B\). You know the basic rule that \(\Pr[A \lor B] = \Pr[A] + \Pr[B] - \Pr[A \land B]\). Just drop the last term, and you have \(\Pr[A \lor B] \leq \Pr[A] + \Pr[B]\). You can apply this iteratively when more than two events are involved.
that recognizes SAT with two-sided error. We will design a randomized, polynomial time algorithm $A$ that recognizes SAT with one-sided error-probability $1/2$.

Let us first discuss the ideas. We need an algorithm $A$ that, unlike $B$, never accepts $\langle \varphi \rangle$ in the case that $\varphi$ is unsatisfiable. To ensure this, $A$, on input $\langle \varphi \rangle$, will seek a “proof” or “certificate” that $\varphi$ is satisfiable, and reject unless it finds one. It will do this by using the reduction of decision to search of SAT described in the Notes on Decision versus Search, with $B$ playing the role of the “magic box.” We now proceed to the actual proof.

Let $\varphi$ be a formula on variables $x_1, \ldots, x_n$. For bits $b_1, \ldots, b_i \in \{0, 1\}$ — think of them as assignments to $x_1, \ldots, x_i$ — let $\varphi_{b_1, \ldots, b_i}$ be the formula on variables $x_{i+1}, \ldots, x_n$ obtained by plugging in $x_1 = b_1, \ldots, x_i = b_i$ in $\varphi$. We let $r$ be the number of coins used by $B$. Then we define the algorithm $A$ as follows:

Algorithm $A(\langle \varphi \rangle; R)$

Let $n$ be the number of variables in $\varphi$

$k \leftarrow 1 + \lceil \log_2(n + 1) \rceil$

Let $R_0$ be the first $r(|\langle \varphi, 1^k \rangle|)$ bits of $R$

If $B(\langle \varphi, 1^k \rangle; R_0) = 0$ then reject

For $i = 1, \ldots, n$:

Let $R_i$ be the next $r(|\langle \varphi_{b_1, \ldots, b_{i-1}, 0, 1^k} \rangle|)$ bits of $R$

If $B(\langle \varphi_{b_1, \ldots, b_{i-1}, 0, 1^k} \rangle; R_i) = 1$ then let $b_i = 0$ else let $b_i = 1$

End for

If $\varphi(b_1, \ldots, b_n) = 1$ then accept else reject.

Here $A$, on input $\langle \varphi \rangle$, is dividing its random tape $R$ into $n + 1$ pieces. It invokes $B$ a total of $n + 1$ times, each time using a different piece $R_i$ of its random tape.

It is clear that if $\varphi$ is unsatisfiable then $A$ will never accept, because, as indicated above, in this case there is simply no sequence $b_1, \ldots, b_n$ for which $\varphi(b_1, \ldots, b_n) = 1$. Now the question is with what probability $A$ accepts when $\varphi$ is satisfiable.

$A$ invokes $B$ a total of $n + 1$ times. If $\varphi$ is satisfiable and each of these invocations returns the correct answer then it ends up with a satisfying assignment. We now show that the probability that some invocation (ie. at least one of the $n + 1$ invocations) returns the wrong answer is at most $1/2$.

The probability that any invocation of $B$ returns the wrong answer is at most $2^{-k}$. Now the probability that the whole procedure returns the wrong answer is at most $(n + 1) \cdot 2^{-k}$ which is at most $1/2$ by the choice of $k$.

2. [10 points] If $\text{NP} \subseteq \text{BPP}$ then $\text{NP} = \text{RP}$.

We know that $\text{RP} \subseteq \text{NP}$. So it suffices to show that if $\text{NP} \subseteq \text{BPP}$ then $\text{NP} \subseteq \text{RP}$.

Assume $\text{NP} \subseteq \text{BPP}$. This means $\text{SAT} \in \text{BPP}$. By part 1. above, $\text{SAT} \in \text{RP}$. Since $\text{SAT}$ is $\text{NP}$-complete, it follows that $\text{NP} \subseteq \text{RP}$.