11.5.12

(a)

Show that it is undecidable to answer the following problem called **RP**-ACCEPTANCE: Given a precise non-deterministic Turing machine (PNTM) M, is it true that for all inputs either all computations reject or at least half of them accept?

Reduction from HALTING. Assume a deterministic Turing machine M' and an input x. Build machine $\hat{M}$ such that, for input y, (i) $\hat{M}$ first builds a yard stick $\sqcap^{|y|}$ on an extra string, (ii) then prints x (which is a constant for $\hat{M}$) on the input string and (iii) finally non-deterministically initiates three deterministic computations of length $|y|$ (the yard stick is used) such that two of them do nothing halting on "no" and one of them executes M' on x for $|y|$ steps and halts on "yes" if M' halted and on "no" otherwise. Now if M' halts on x then there is an input y of sufficient length such that one computation of $\hat{M}$ accepts and two of them do not. In that case, $\hat{M}$ does not fulfill the **RP** accepting condition. On the other hand, if M' does not halt, then all three computations of $\hat{M}$ always reject for all inputs y and $\hat{M}$ fulfills the **RP** accepting condition. Therefore, if the **RP**-ACCEPTANCE were decidable, then HALTING would be, meaning that **RP**-ACCEPTANCE is undecidable.

(b)

Show that it is undecidable to answer the following problem called **BPP**-ACCEPTANCE: Given a precise non-deterministic Turing machine (PNTM) M, is it true that for all inputs either at least $\frac{3}{4}$ of the computations reject or at least $\frac{3}{4}$ of them accept?

Reduction from HALTING. Assume a deterministic Turing machine M' and an input x. Build machine $\hat{M}$ such that, for input y, (i) $\hat{M}$ first builds a yard stick $\sqcap^{|y|}$ on an extra string, (ii) then prints x on the input string and (iii) finally non-deterministically initiates two deterministic computations of length $|y|$ (the yard stick is used) such that two of them do nothing halting on "no" and one of them executes M' on x for $|y|$ steps and halts on "yes" if M' halted and on "no" otherwise. Now if M' halts on x then there is an input y of sufficient length such that $\frac{1}{2}$ of the computations of $\hat{M}$ accept. In that case, $\hat{M}$ does not fulfill the **BPP** accepting condition. On the other hand, if M' does not halt on x, then all three computations of $\hat{M}$ always reject for all inputs y and $\hat{M}$ fulfills the **RP** accepting condition. Therefore, if **BPP**-ACCEPTANCE were decidable, then HALTING would be, meaning that **BPP**-ACCEPTANCE is undecidable.

(c)

Show that it is undecidable to answer the following problem called AT LEAST ONE ACCEPTS: Given a precise non-deterministic Turing machine (PNTM) M, is it true that for all inputs at least one computation accepts?

Reduction from HALTING. Assume a deterministic Turing machine M' and an input x. Build machine $\hat{M}$ such that, for input y, (i) $\hat{M}$ first builds a yard stick $\sqcap^{|y|}$ on an extra string, (ii) then prints x on the input string and (iii) executes M' on x for $|y|$ steps and rejects if M' halted and accepts otherwise. Now if M' halts on x then there is an input y of sufficient length such that $\hat{M}$ rejects (on the single computation). In that case, the answer to AT LEAST ONE ACCEPTS on $\hat{M}$ is "no". On the other hand, if M' does not halt on x, then all computations of $\hat{M}$ always accept for all inputs y and the answer to AT LEAST ONE ACCEPTS on $\hat{M}$ is "yes". Therefore, if AT LEAST ONE ACCEPTS were decidable, then HALTING would be, meaning that AT LEAST ONE ACCEPTS is undecidable.

(d)

Show that it is undecidable to answer the following problem called XOR ACCEPTING: Given a two non-deterministic Turing machines (PNTM), M_1 and M_2, is it true that for all inputs either the first has an accepting condition, or the second does, but never both?

Reduction from HALTING. Assume a deterministic Turing machine M' and an input x. Build machine $\hat{M}_1$ such that, for input y, (i) $\hat{M}_1$ first builds a yard stick $\sqcap^{|y|}$ on an extra string, (ii) then prints x on the input string and (iii) executes M' on x for $|y|$ steps and accepts if M' halted and rejects otherwise. Build machine $\hat{M}_2$ such that, for input y, (i) $\hat{M}_2$ first builds a yard stick $\sqcap^{|y|}$ on an extra string, (ii) then prints x on the input string and (iii) does nothing for $|y|$ steps and accepts. Now if M' halts on x then there is an input y of sufficient length such that $\hat{M}_1$ and $\hat{M}_2$ both accept. In that case, the answer to XOR ACCEPTING on $\hat{M}_1$ and $\hat{M}_2$ is "no". On the other hand, if M' does not halt on x, the $\hat{M}_1$ always rejects and $\hat{M}_2$ always accepts and and the answer to XOR ACCEPTING is "yes". Therefore, if XOR ACCEPTING were decidable, then HALTING would be, meaning that XOR ACCEPTING is undecidable.

11.5.13

Déjà vu: A class $\mathcal{C}$ is *closed under reductions* if it holds that whenever a language L reduces to a language $L' \in \mathcal{C}$, then $L \in \mathcal{C}$.

The problem is to show that the classes **RP**, **BPP** and **PP** are closed under reductions.

Proof by hand-waving. Note that each class in question can also perform deterministic computation as explained in the book. Now assume that a reduction from a language L exists to a language $L' \in \mathcal{C}$ where $\mathcal{C} \in \{\mathbf{RP}, \mathbf{BPP}, \mathbf{PP}\}$. Clearly $L \in \mathcal{C}$ since it is decided by the following precise machine M fulfilling the accepting condition required for $\mathcal{C}$: M first performs the (log-space and therefore polynomial length) deterministic reduction function (which was assumed to exists) on the input x and then executes the $\mathcal{C}$-machine M' on the transformed input, deciding L' (one has to exist since $L' \in \mathcal{C}$). Clearly M is a $\mathcal{C}$-machine and decides L by the definition of reductions.

11.5.15

The problem is to show that **PP** is closed under complement and symmetric difference.

Complement

A class $\mathcal{C}$ of languages is closed under complement if it holds that: for all languages $L \in \mathcal{C}$, the complement $\overline{L}$ of L is in $\mathcal{C}$.

Consider a precise non-deterministic polynomial-time Turing machine M that **PP**-decides L. Let $p(n)$ be the precise polynomial bound of M. Build a precise non-deterministic polynomial-time Turing machine M' as follows. On input x M first computes the yard stick $p(|x|)$ on an extra string. M then non-deterministically does one of the following: (i) it executes M on x (which takes time precisely $p(|x|)$) and rejects if M accepts and vice versa or (ii) it executes a deterministic computation of length $|x|$ by using the yard stick and accepts. Now if $x \in L$, then more than half of the computations of M accept x and thus "less than half + 1 = at most half" of the computations of M' accepts x. On the other hand, if $x \notin L$, then at least half of the computations of M reject x and thus "at least halt + 1 = more than half" of the computations of M' accept x. Therefore M' **PP**-decides $\overline{L}$ and **PP** is closed under complement.

Symmetric Difference

A class $\mathcal{C}$ of languages is closed under symmetric difference if it holds that for all languages $L_1, L_2 \in \mathcal{C}$, the language $(L_1 \setminus L_2) \cup (L_2 \setminus L_1)$ is in $\mathcal{C}$.

Let M_1 and M_2 be the precise nondeterministic Turing machines that **PP**-accept L_1 and L_2, respectively. Now consider the machine M that first runs M_1 on input x and then runs M_2 on the same input and then accepts if either but not both of the machines M_1, M_2 accepted. Let p_1 be the "probability" of an accepting computation of M_1 on x i.e. the ration of accepting computations and the number of computations of M_1 on x. Similarly p_2 for M_2 on x.

Now if $x \notin L_1$ and $x \notin L_2$, then the worst case is when exactly half of the computations of the first machine accept. But then the probability of M accepting is at most $\frac{1}{2} \times (1-p_2) + p_2 \times (1 - \frac{1}{2}) = \frac{1}{2}$ and thus M rejects x which is correct.

If $x \in L_1$ and $x \notin L_2$, then $p_1 \times (1-p_2) + p_2 \times (1-p_1)$ is minimum when $p_2 = 0$ and thus the probability of M accepting is p_1 which is more than half and correct. Symmetric argumentation holds for the case $x \notin L_1$ and $x \in L_2$.

Finally, if $x \in L_1$ and $x \in L_2$, then $p_1 \times (1 - p_2) + p_2 \times (1-p_1)$ is maximum when $p_1 = \frac{1}{2} + \epsilon$, $\epsilon > 0$, and thus the probability of M accepting is $\frac{1}{2} + \epsilon - \frac{1}{2} \times p_2 - \epsilon \times p_2 + p_2 - p_2 \times (\frac{1}{2} + \epsilon) = \frac{1}{2} + \epsilon + p_2 - p_2 - p_2 \times 2 \times \epsilon = \frac{1}{2} + \epsilon \times (1 - 2 \times p_2)$ which in turn is maximum when $p_2 = \frac{1}{2} + \epsilon'$, $\epsilon' > 0$, and thus the probability of M accepting is $\frac{1}{2} - \epsilon \times 2 \times \epsilon'$ which is less than $\frac{1}{2}$ and therefore correct.

To sum up, we have shown that M **PP** accepts the symmetric difference of L_1 and L_2 and therefore **PP** is closed under symmetric differences.