

1. Problem 11.5.17 of the text.

Let $0 < \epsilon < 1$ be a rational number. We say that $L \in \mathbf{PP}_\epsilon$ if there is a nondeterministic Turing machine M such that $x \in L$ if and only if at least an ϵ fraction of the computations are accepting. Show that $\mathbf{PP}_\epsilon = \mathbf{PP}$.

2. Problem 11.5.18 of the text.

Show that, if $\mathbf{NP} \subseteq \mathbf{BPP}$, then $\mathbf{RP} = \mathbf{NP}$. (That is, if \mathbf{SAT} can be solved by randomized machines, then it can be solved by randomized machines with no false positives, presumably by computing a satisfying truth assignment as in Example 10.3.)

3. Problem 11.5.25 of the text.

We know that most languages do not have polynomial circuits (Theorem 4.3), but that certain undecidable ones do (Proposition 11.2). We suspect that \mathbf{NP} -complete languages have no polynomial circuits (Conjecture B in Section 11.4). How high do we have to go in complexity to find languages that *provably* do not have polynomial circuits?
