The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Solution Manual
Chapter 8, Additional Problem 8.8

In other words, if *SAT* (or any other NP-complete problem) is solvable at all in randomized polynomial time, it is solvable in randomized polynomial time with one-sided error.

In order to ensure that only "yes" instances can be accepted, we shall resort to testing a truth assignment and accept only if the truth assignment satisfies the instance. In order to create the truth assignment to be tested (which has to be the "right" assignment with probability at least ε), we shall use the BPP algorithm to test the assignment of a truth value to each variable in turn.

Let our BPP solution algorithm for *SAT* be $\mathcal{A}$. This algorithm returns "yes" on "yes" instances with probability at least $1-\delta$ and returns "no" on "no" instances with probability at least $1-\delta$; by Theorem 8.29, we can make the δ any chosen constant by running the basic BPP algorithm $\Theta(\log \delta)$ times. In particular, we can run the algorithm on input x a linear number of times and thus make δ as small 2^{-n}, where n is the number of variables in instance x. Such an iterated algorithm, call it $\mathcal{B}$, when working on a "yes" instance, will reject it with probability at most 2^{-n} and, when working on a "no" instance, will accept it with probability at most 2^{-n}.

Now consider some unknown instance x; we shall use algorithm $\mathcal{B}$ to attempt to discover a satisfying truth assignment for x, accepting x only if such an assignment is found. Clearly, such a procedure will reject all "no" instances—we just need to ensure that it runs in polynomial time and accepts a "yes" instance with some minimum probability ε. We consider each variable in turn. Say we tentatively set x_i to true; we reflect that assignment in the instance x producing the new instance $x|_{x_i=1}$ and run algorithm $\mathcal{B}$ on $x|_{x_i=1}$. If the algorithm answers "yes," we permanently set x_1 to true, otherwise we permanently set it to false. Now we proceed the same way with the next variable, until we have assigned all variables. Once all variables are set, we check the resulting truth assignment and accept x if and only if that assignment satisfies x. This last step ensures that all "no" instances are always rejected.

Let us now compute the probability that the truth assignment derived in our procedure is a satisfying truth assignment when x is a "yes" instance. If x is a "yes" instance, then it has at least one certificate—one satisfying truth assignment. In the worst case, it has only one such assignment. In that case, our procedure assigns the wrong value to the first variable with probability at most $2 \cdot 2^{-n} = 2^{1-n}$; that is, it succeeds with probability at least $1 - 2^{1-n}$. The same is true at every step, since the number of iterations used by $\mathcal{B}$ is always the same. Thus our procedure succeeds in assigning all n variables with probability at least $(1 - 2^{1-n})^n$, which is at least $1 - n2^{1-n}$ and thus at least as large as $\frac{1}{4}$ for all $n \geq 3$. (For instances of one and two variables, we can trivially exhaust all truth assignments in polynomial time.)