CS5110: Complexity Theory

Shreyas Havaldar
CS18BTECH11042

October 23, 2020

1 Exercise Set 11

1.1 Assuming I have a SAT Oracle, constructing an algorithm that takes input ϕ and outputs a satisfying assignment if $\phi \in SAT$, in poly n time, where n is the input size.

Showing our algorithm running in m steps where m is the number of variables in our CNF. The entire algorithm is poly in n.

1. Query the oracle on the formula ϕ.
2. If the oracle rejected, we can't have a satisfying assignment obviously and so we simply terminate and return false.
3. If the oracle accepted, we're good to go and have a satisfying assignment to be found.
4. We basically run a loop m times and by checking whether a particular assignment to some variable to ϕ still maintains its satisfiability, if it does, we replace the variable with the assignment for further evaluations.
   (a) Let i = 1, and $\phi_0 = \phi$. ϕ_i are just the different iterations of the same CNF formula with i variables assigned a value TRUE or FALSE successively in each ϕ_i.
   (b) For i from 1 to m:
      i. $x_i = 1$, $\phi_i = \phi_{i-1}$ with x_i replaced with value 1 throughout the formula. (x_i is assigned TRUE.)
      ii. Query the oracle on the formula ϕ_i.
      iii. If the oracle accepts, our assignment is one of the possible correct ones, so we note the value of x_i as 1 for further runs of the loop and replace it in the subsequent ϕ_i's, so ϕ_i stays as it is. (As we modified it in step 4.b.i)

Else if the oracle rejected, our assignment cannot be satisfiable with $x_i = 1$ and thus it has to be satisfiable for $x_i = 0$, as the formula is indeed satisfiable and each variable can take only two values, so we note the value of x_i as 0 (x_i is assigned FALSE) for further runs of the loop and replace it in the subsequent ϕ_i's and thus $\phi_i = \phi_{i-1}$ with x_i replaced with 0 throughout the formula.

iv. We also write down the value of x_i (that we replaced throughout the formula above and was thus the assignment for some satisfying assignment), on the i^{th} cell of an output tape.

v. Increment the value of i.

(c) Output the values stored on i^{th} cell of the input tape as the value assigned to the i^{th} variable in some satisfying assignment of the input formula ϕ. At the end of the loop each variable would have been assigned and replaced by a value 0 or 1 in the formula and we would have a satisfying assignment by our algorithm.

5. The time take for the algorithm to run is the substitution of value of every x_i in the formula at most twice, which requires time to traverse the formula and replace the variable with the value so we take atmost 2*$|Inp|$ time per variable, where $|Inp|$ is the size of the input.

6. Querying the oracle with a ϕ_i takes constant time per variable.

7. Where m is the number of input variables we take $O(m * |Inp|)$ time for the algorithm, and as m can be atmost k*C where k is the number of literals in the CNF formula, and C the number of clauses in our CNF, our m is thus $O(|Inp|)$, thus the total algorithm is $O(|Inp|^2)$.

8. Hence we have a satisfying assignment found in poly of input time using just a SAT Oracle!

Hence Constructed!