

Homework 07

April 29, 2020

1. In order for A to be decided, there must be a TM that acts as a decider, and thus, the language is recognizable. Furthermore, we know that $A \leq_M A^C$. Simply switching the membership of these sets so that A now has the elements of A^C will continue to preserve the relationship where all elements on one side of the reduction go to all of the elements not on that side after being reduced, we know $A^C \leq_M A$.
2. As presented in the book, we know the corollary that a language is decidable iff some nondeterministic TM decides it. Therefore, to show that a language is decidable, we can show that you can construct a nondeterministic enumerator to decide the language only if it enumerates in lexicographic order. Let this enumerating decider be D . Consider if D enumerates all decided inputs, up to some infinite amount, in a random order. If D ran on the input, there is no way to decide if the required value will not be present in the language or if it is simply further in the infinite list being produced by the enumerator. Said another way, the decider never knows when it should halt. Alternatively, if the output of the enumerator is in lexicographic order, it is quite simple to halt after it is no longer possible to find the checked string in the enumerated output. Therefore, we can conclude that a language is decidable if and only if an enumerator enumerates in lexicographic order.
3. We can describe an $O(n^k)$ algorithm that searches for cliques of size k in a graph G :
 - For each edge v in G
 - – For each edge e connected to v
 - * "explore" the other vertex u in edge e
 - * reduce k by 1
 - * repeat until $k = 0$
 - * check if the explored nodes form a clique. If they do, stop. If not, continue

This does not imply $CLIQUE \in P$ because this solution is still one that exhaustively searches through every solution in an exponential time complexity. The only way for it to be reduced to a polynomial time is through nondeterministic search options.

4. First, we can begin with a clique problem to find a clique of size k on graph G_1 . From there, we know that the only way G_2 can also have a clique of size k is if there is a subgraph in G_1 that is isomorphic to G_2 . Therefore, we can reduce the clique problem to the subgraph isomorphism problem by running the clique problem on both graphs since the only way for them to produce a clique of the same size k is if there is an isomorphism from G_1 to G_2 .

5. Showing that (b) and (c) are equivalent is pretty trivial. By the definition of an NP complete problem, one can be reduced to any other NP complete problem. Therefore, if every NP complete problem is in P, then there is always some problem that is NP complete that is in P, and if one NP complete problem is in P, then all NP complete problems are in P. Furthermore, if every NP complete problem is in P, that means every problem that could non-nondeterministically be computed in P time can also be computed in deterministic time. Therefore since all the NP problems could be reduced to P problems, $P = NP$.