Assignment 6, CS 6363

due at the beginning of Lecture on 11/21

no late homework would be accepted

Homework consists of the following 7 problems:

1 (Exercise 34.1-6) Show that the class \( P \), viewed as a class of languages, is closed under union, intersection, concatenation, complement, and Kleene closure. That is, if \( A, B \in P \), then \( A \cup B \in P, \ A \cap B \in P, \ AB \in P, \ \overline{A} \in P \) and \( A^* \in P \).

Solution Since \( A, B \in P \), checking whether \( x \in A \) can be done in polynomial-time and checking whether \( x \in B \) can also be done in polynomial-time. The following are polynomial-time algorithms for checking \( x \in A \cup B, x \in A \cap B, x \in AB, x \in \overline{A} \) and \( x \in A^* \).

Algorithm Union

input \( x \);

if \( x \in A \) or \( x \in B \) then accept

else reject.

Algorithm Intersection

input \( x \);

if \( x \in A \) and \( x \in B \) then accept

else reject.

Algorithm Concatenation

input \( x \);

\( ok := 0 \);

for each pair of strings \( a \) and \( b \) satisfying \( x = ab \) do

\( \text{if } a \in A \) and \( b \in B \) then \( ok := 1; \)
if \( ok = 1 \) then accept
    else reject.

Algorithm Complement
input \( x \);
if \( x \in A \) then reject
    else accept.

Algorithm Star Closure
input \( x \);
\( ok := 0 \);
for each pair of nonempty strings \( a \) and \( b \) satisfying \( x = ab \) do
    if \( a \in A \) and \( b \in A^* \) then \( ok := 1 \);
if \( ok = 1 \) then accept
    else reject.

\( \square \)

2 (Exercise 34.2-4) Show that the class \( NP \) is closed under union, intersection, concatenation, and Kleene closure. Discuss the closure of \( NP \) under complement.

Solution Since \( A, B \in NP \), checking whether \( x \in A \) can be done in nondeterministic polynomial-time and checking whether \( x \in B \) can also be done in nondeterministic polynomial-time. The following are nondeterministic polynomial-time algorithms for checking \( x \in A \cup B, x \in A \cap B, x \in AB, \) and \( x \in A^* \).

Algorithm Union
input \( x \);
if \( x \in A \) or \( x \in B \) then accept.
Algorithm Intersection
input $x$;
if $x \in A$ and $x \in B$ then accept.

Algorithm Concatenation
input $x$;
guess a pair of strings $a$ and $b$ satisfying $x = ab$;
if $a \in A$ and $b \in B$ then accept.

Algorithm Star Closure
input $x$;
guess an integer $k$, $1 \leq k \leq |x|$; guess $k$ strings $a_1, \ldots, a_k$ with length at most $|x|$;
if $x = a_1 a_2 \cdots a_k$
  if $a_1 \in A$, $a_2 \in A$, $\ldots$, $a_k \in A$
    then accept.

It is not know whether $NP = co - NP$, i.e., it is an open problem whether complement operation is closed in NP. \hfill $\square$

3 (Exercise 34.4-5) Show that the problem of determining the satisfiability of Boolean formula in disjunctive normal form is polynomial-time solvable.

Solution Given a DNF $F$, we check each elementary product $P$ in $F$. $F$ is satisfiable if and only if there exists an elementary product $P$ such that for every variable $x_i$, either $x_i$ or $\bar{x}_i$ does not appear in $P$. This can be done in polynomial-time. \hfill $\square$

4 (Exercise 34.5-1) The subgraph-isomorphism problem takes two graphs $G_1$ and $G_2$ and asks whether $G_1$ is isomorphic to a subgraph of $G_2$. Show that the subgraph-isomorphism problem is NP-complete.

Solution We reduce Hamiltonian Cycle problem to this problem. For each input graph $G = (V, E)$ of Hamiltonian Cycle problem, we construct two graphs $G_1$ and $G_2$ as follows: $G_1$
is the cycle \((v_1, v_2, \ldots, v_n)\) where \(V = \{v_1, v_2, \ldots, v_n\}\). \(G_2 = G\). Clearly, \(G\) has a Hamiltonian cycle if and only if \(G_1\) is isomorphic to a subgraph of \(G_2\). \(\Box\)

5 (Exercise 34.5-7) The longest-simple-cycle problem is the problem of determining a simple cycle (no repeated vertices) of maximum length in a graph. Show that this problem is NP-hard.

**Solution** The decision version of the longest-simple-cycle problem can be stated as follows: Given a graph \(G\) and a positive integer \(k\), determine whether \(G\) has a cycle with length at least \(k\). We reduce the Hamiltonian Cycle problem to it. For input graph \(G = (V, E)\) of the Hamiltonian Cycle problem, we construct input \(<G, k>\) of the decision version of the longest-simple-cycle problem by setting \(k = |V|\). Then \(G\) has a Hamiltonian cycle if and only if \(G\) has a simple cycle with length at least \(k\). \(\Box\)

6 Show that the following problem is NP-complete: Given a graph \(G\), determine whether \(G\) contains a Hamiltonian path where a path is Hamiltonian if it passes every vertex exactly once.

**Solution** First, show that the Hamiltonian Path problem belongs to NP. To do so, for input graph \(G = (V, E)\), guess an ordering of vertices and check if this ordering gives a Hamiltonian path. If yes, then accept. This gives nondeterministic polynomial-time algorithm for the Hamiltonian Path problem. Therefore, the Hamiltonian Path problem belongs to NP.

We next reduce the Hamiltonian Cycle problem to the Hamiltonian Path problem as follows: For input graph \(G = (V, E)\) of the Hamiltonian Cycle problem, consider a vertex \(v \in V\) and all its neighbors \(v_1, v_2, \ldots, v_d\). Let \(G'\) be a graph obtained from \(G\) by adding three new vertex \(u', w\) and \(w'\), and adding new edges \((u, w), (u', w'), (u', v_1), \ldots, (u', v_d)\). Then \(G\) contains a Hamiltonian cycle if and only if \(G'\) contains a Hamiltonian path. (Prove it!!!) \(\Box\)

7 Show that the following problem is NP-hard: Given a graph, find a spanning tree to maximize the number of leaves.

**Solution** The decision version of this problem is as follows: Given a graph \(G = (V, E)\) and a
positive integer $k$, determine whether $G$ has a spanning tree with at least $k$ leaves.

We reduce the Vertex-Cover problem to it. For input graph $G = (V, E)$ and positive integer $h$ of the Vertex-Cover problem, we construct an input $G' = (V', E'), k > 0$ of the above problem in the following way: Let $V' = V \cup E \cup \{s, t\}$ where $s$ and $t$ are two vertices not in $V \cup E$. Let $E' = \{(u, e), (v, e) \mid e = (u, v) \in E\} \cup \{(s, t)\} \cup \{(s, u) \mid u \in V\}$. Set $k = 1 + |E| + |V| - h$.

First, assume $G$ has a vertex cover $C$ of size at most $h$. Then $G'$ has a spanning tree with at least $k$ leaves, which consisting of edge set $\{(s, t)\} \cup \{(s, u) \mid u \in V\} \cup F$ where $F$ is a subset of $\{(u, e), (v, e) \mid e = (u, v) \in C\}$ such that for every $e \in E$, there is exactly one $u \in C$ such that $(u, e) \in F$.

Next, assume $G'$ has a spanning tree $T$ with at least $k$ leaves. Without loss of generality, we may assume that $(s, u) \in T$ for every $u \in V$. In fact, if $(s, u) \notin T$, then we must have $(u, e) \in T$ for some $e \in E$. Replacing $(u, e)$ by $(s, u)$, we will obtain a spanning tree also with at least $k$ leaves. Now, every vertex $e \in E$ must be a leaf since, otherwise, $T$ would not be a tree. Set $C = \{u \in V \mid u$ is not a leaf of $T\}$. Then $C$ is a vertex cover and $|C| \leq 1 + |V| + |E| - k = h$.

\[ \square \]

**The following are recommended exercises**

8 Given a graph, find the maximum independent set. (An independent set is a subset of vertices which are not adjacent each other.)

9 Given a weighted complete graph, find a maximum weight tour. (A tour is a hamiltonian cycle.)

10 Given a graph $G = (V, E)$, compute its minimum connected dominating set where a vertex subset $C$ is called a connected dominating set if every vertex is either in $C$ or adjacent to a vertex in $C$, and in addition the subgraph induced by $C$ is connected.

11 (Problem 34-2a) Bonnie and Clyde have just robbed a bank. They have a bag of money and want to divide it up. For each of the following scenarios, either give a polynomial-time
algorithm, or prove that the problem is NP-complete. The input in each case is a list of the $n$ items in the bag, along with the value of each. (a) The bag contains $n$ coins, but only 2 different denominations: some coins are worth $x$ dollars, and some are worth $y$ dollars. Bonnie and Clyde wish to divide the money exactly evenly.

12 (Problem 34-2b) The bag contains $n$ coins, with an arbitrary number of different number of different denominations, but each denomination is a nonnegative integer power of 2, i.e., the possible denominations are 1 dollar, 2 dollars, 4 dollars, etc. Bonnie and Clyde wish to divide the money exactly evenly.

13 Given a cubic graph, find the minimum vertex cover. (A cubic graph is a graph such that every vertex has degree three.)

14 Given a set $X$ and $m$ subsets $X_1, X_2, \ldots, X_m$ that $|X_i| \leq 3$ for all $i$, find the a graph $G$ with the minimum number of edges such that for every $i$ $G$ contains a spanning tree for $X_i$.

15 Given a graph $G$, determine whether $G$ has a vertex cover $C$ satisfying the following conditions:

(a) The subgraph $G|_C$ induced by $C$ has no isolated point.

(b) Every vertex in $C$ is adjacent to a vertex not in $C$.

16 There are $n$ students who studied at a late-night study for final exam. The time has come to order pizzas. Each student has his own list of required toppings (e.g. mushroom, pepperoni, onions, garlic, sausage, etc). Everyone wants to eat at least one third of a pizza, and the topping of that pizza must be in his required list. A pizza may have only one topping. Find the minimum number of pizzas to order to make everyone happy.

17 Show the NP-hardness of the following problem: Given a directed graph, find the minimum subset of vertices such that every directed cycle contains at least one vertex in the subset.