Homework 5

Rebah Özkoç

0. **Suppose M' is a TM that semidecides a language L. Construct a TM M making use of M' that semidecides the language L*.**

Solution:

To semidecide the language L*, which represents the Kleene closure of L, we can construct a Turing Machine M using M' which semidecides L. L * is the set of all strings obtained by concatenating zero or more strings from L. M takes an input w and semidecides if w Є L*. I assumed that there is enough tapes to store the all of the partitions of w. This multi-tape TM operates like this:

1. Put the initial input w on tape 1.
2. Crate all possible partitions of w recursively.
3. Copy all the partitions of w into different tapes.
4. Run the Turing machine M on each part of the partition in parallel.
5. If all of M's that are working on the parts of a partition halts this means all the parts of the partition belong to L. In this case halt the M'.

1. **Prove the transitivity of the polynomial reduction operator α : i.e. , L1α L2 and L2 α L3 implies that L1 α L3**

Solution:

If a language L1 is polynomially reducible to another language L2, it means that we can solve L1 in polynomial time using a procedure that solves L2, implying that L1 is no harder than L2. According to the definition, there exist poly-time functions f and g such that x Є L1 ⟷ f(x) Є L2 and y Є L2 ⟷ g(y) Є L3, thus x Є L1 ⟷ f(x) Є L2 ⟷ g(f(x)) Є L3. |f(x)| is polynomial in x, thus g(f(.)) is a poly-time function by the definition of composite functions.

2. **Given a SAT problem define a set of literals in SAT a consistent set if a literal xj and its complement literal x c j are NOT both members of this set. Prove that SAT has a solution if and only if there exists a consistent set of literals, whose members are selected, one from each clause Cj.**

Solution:

The SAT problem is a NP-Complete decision problem that takes as input a Boolean expression and asks whether there is some assignment of true and false values to the variables in the expression such that the entire expression evaluates to true.

Now we can consider a SAT problem with clauses C1, C2, ..., Cn. Each clause is a disjunction (or) of literals (either a variable or its complement). We define a set of literals as consistent if a literal xj and its complement literal ¬xj are NOT both members of this set.

For example:

{x1, x2, ¬x3} is a consistent set of literals but {x1, x3, ¬x3} is not a consistent set of literals.

To prove the statement, we need to prove the two directions of the statement:

If SAT has a solution, then there exists a consistent set of literals, whose members are selected, one from each clause.

If there exists a consistent set of literals, whose members are selected, one from each clause, then SAT has a solution.

Direction 1:

If SAT has a solution, then there exists a consistent set of literals, whose members are selected, one from each clause.

Proof: Suppose SAT has a solution. This means there is an assignment of variables that makes the entire statement true. For each clause Cj, pick a literal that is true under this assignment (there must be at least one, otherwise the clause would not be satisfied). The selected literals form a set. It is consistent by definition because no variable and its complement can be true at the same time.

Direction 2: If there exists a consistent set of literals, whose members are selected, one from each clause, then SAT has a solution.

Proof: Suppose there exists a consistent set of literals, whose members are selected, one from each clause. We can create an assignment based on this set by assigning TRUE to each literal in the set and FALSE to its complement. This assignment satisfies every clause because for each clause, we've selected a literal from it that is assigned TRUE. Hence, this assignment satisfies the entire formula, which means SAT has a solution.

3. Prove the following: IS α CLIQUE , IS α NC, SAT α MAXSAT, HC α UHC where IS= Independent Set , NC = Node Cover and UHC = HC for undirected graphs

Solution:

Independent Set Problem is reducible to CLIQUE Problem:

We can take the complement of the graph G, which we'll call G'. The complement of a graph G is a graph on the same vertices such that two distinct vertices of G' are adjacent if and only if they are not adjacent in G. In other words, an edge exists between two vertices in G' if and only if the edge does not exist between those two vertices in G.

Now, we can observe that any independent set S in G forms a clique in G', and any clique in G' forms an independent set in G. This is because by definition, no two vertices in S share an edge in G. Therefore, in G', all vertices in S are connected.

Thus, if we can solve the Clique Problem in G', we can solve the Independent Set Problem in G. We just need to convert G into G' and apply a solver for the Clique Problem.

This is a polynomial time reduction from Independent Set to Clique Problem because creating the complement of a graph can be done in polynomial time. Therefore, we can say that Independent Set Problem is reducible to the Clique Problem.

Independent Set Problem is reducible to Node Cover Problem:

To prove that Independent Set (IS) is reducible to Node Cover (NC), we must show that for any instance of the IS problem, we can construct an instance of the NC problem, such that the solution to the NC problem can be used to find a solution to the original IS problem.

Given an instance of IS on a graph G=(V, E), where V is the set of vertices and E is the set of edges, we construct an instance of NC on the same graph G. The graph remains the same, so this transformation can be done in polynomial time.

Now, observe that a set of nodes S is a node cover in G if and only if the set of nodes V-S (i.e., all nodes not in S) is an independent set in G. This is because S is a node cover means every edge in G has at least one end in S, hence V-S can't have any edge among its nodes, making it an independent set.

This means that finding a minimum node cover in G is equivalent to finding a maximum independent set in G. We can convert the solution to the NC problem back to a solution to the IS problem by simply taking the complement of the node cover (i.e., all nodes not in the node cover).

Hence, we have shown a polynomial time reduction from IS to NC, proving that IS Problem is reducible to NC Problem. Therefore, any algorithm that solves the NC problem can be used to solve the IS problem, implying that the IS problem is not harder than the NC problem.

SAT Problem is reducible to MAXSAT Problem:

If we can reduce SAT to MAXSAT, then we can use a solution for MAXSAT to solve SAT.

The idea is to encode a SAT problem into a MAXSAT problem in polynomial time. This can be done quite straightforwardly.

Given a Boolean formula F with m clauses for the SAT problem, we create a MAXSAT problem with the same Boolean formula and ask whether there exists an interpretation that can satisfy m clauses.

Here's how the reduction works:

Input: A Boolean formula F (with m clauses) for SAT.

Output: The same Boolean formula F for MAXSAT with the number of clauses to satisfy as m.

This reduction is correct because if there is a solution for the MAXSAT problem, then that solution would also satisfy the SAT problem. This solution satisfies m clauses, which is exactly all the clauses of the Boolean formula F, and hence, is a satisfying assignment for SAT.

Therefore, we have proved that SAT is reducible to MAXSAT in polynomial time, which means that SAT is no harder than MAXSAT in terms of computational complexity.

Hamiltonian Cycle Problem is reducible to Undirected Hamiltonian Cycle Problem:

To reduce Hamiltonian Cycle (HC) to Undirected Hamiltonian Cycle (UHC), we need to transform each directed edge in the directed graph into undirected edges such that the undirected graph maintains the properties of the original directed graph.

Consider a directed graph G=(V,E) with vertices V and edges E.

The transformation works as follows:

For each directed edge (u,v) in E, replace it with an intermediary node x and two undirected edges (u,x) and (x,v).

This transforms the directed graph into an undirected graph G'=(V',E') with vertices V' and edges E' while preserving the same Hamiltonian cycles as G.

Any Hamiltonian cycle in G must pass through each vertex once, and so must pass through each edge once. In the transformed graph, this corresponds to passing through each pair of edges (u,x) and (x,v) in G' exactly once, which is the same as passing through the intermediary node x once. Hence, any Hamiltonian cycle in G corresponds to a Hamiltonian cycle in G'.

Any Hamiltonian cycle in G' must pass through each intermediary node x once. Because x is only connected to two nodes u and v, this means the cycle must also pass through (u,x) and (x,v) exactly once. Hence, any Hamiltonian cycle in G' corresponds to a Hamiltonian cycle in G.

In this way, the problem of finding a Hamiltonian cycle in G has been reduced to the problem of finding a Hamiltonian cycle in G', proving that HC ≤ UHC.

This reduction can be computed in polynomial time because for each edge in G, a constant number of operations are required to create G', which means the reduction is computable in O(n) time where n is the number of edges in G. Hence, this is a polynomial time reduction.

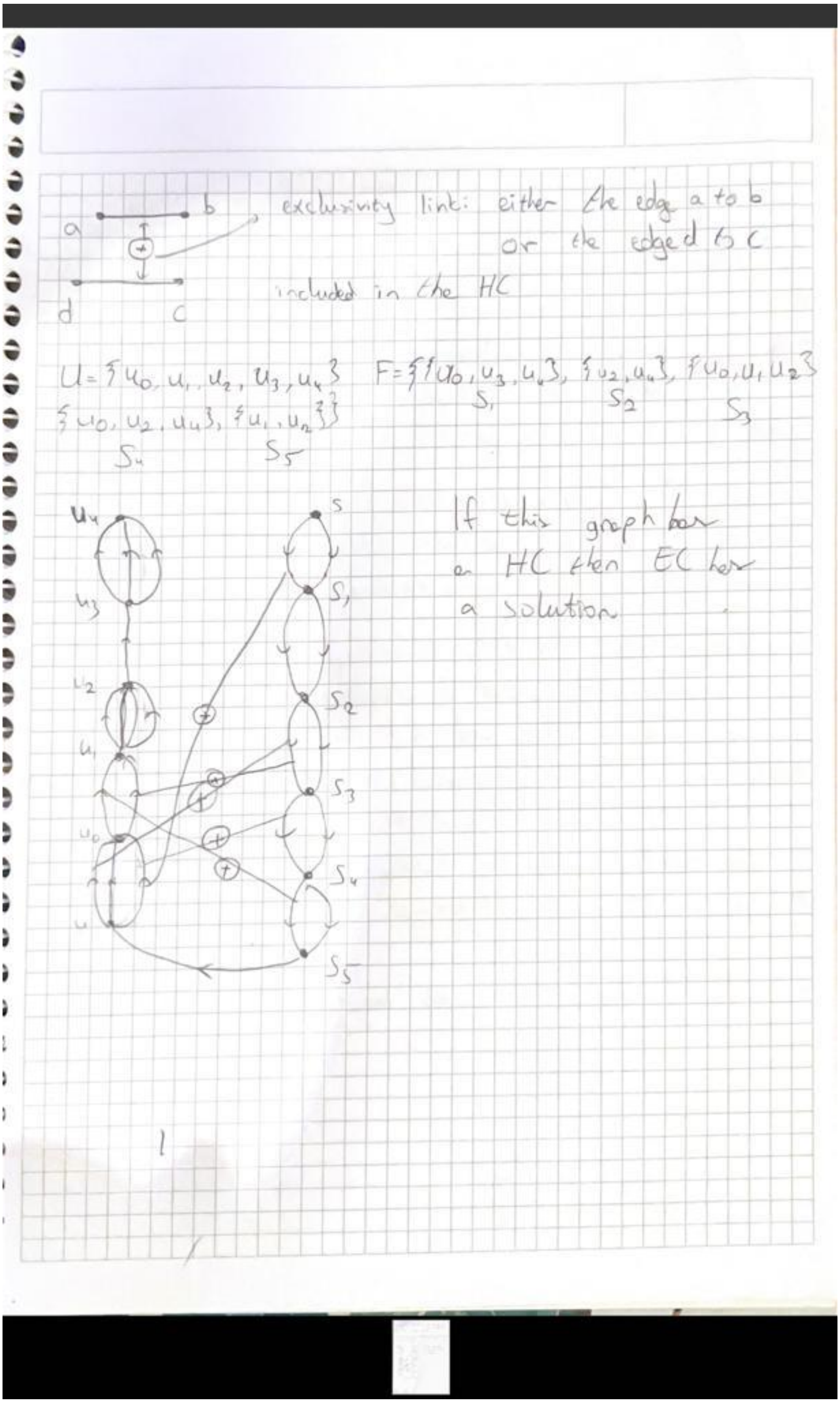
4. a) Formulate the 2SAT problem where each vertex corresponds to a Boolean literal and there is a directed edge from vertex x to vertex y corresponding to x implies y (x ⇒ y or ¬x ∨ y or (¬x, y) is a clause)
   b) Show that 2SAT ∈ P

Solution:

5. Given an EC problem with U = { u0, u1 , u2 , u3 , u4 } ; F = { {u0 , u3 , u4}, {u2 ,u4}, {u0 , u1, u2}, {u0, u2, u4}, {u1, u2} } State the KS and the HC problems obtained from the above EC problem by the polynomial reduction methods discussed in class. State solution(s) of the three problems EC , KS and HC if one exists for each case.

Solution:

HC Problem that is obtained from this EC problem:



a, b, d, c

exclusivity link: either the edge a to b or the edge d to c included in the HC

$U = \{u_0, u_1, u_2, u_3, u_4\}$ $F = \{\{u_0, u_3, u_4\}_{S_1}, \{u_2, u_4\}_{S_2}, \{u_0, u_1, u_2\}_{S_3}, \{u_0, u_2, u_4\}_{S_4}, \{u_1, u_2\}_{S_5}\}$

If this graph has a HC then EC has a solution