CS 4510: Automata and Complexity **Fall 2019**

Lecture 18: Reduction

November 6, 2019

Lecturer: Santosh Vempala *Scribe: Nicolas Soong*

18.1 Introduction

Suppose A and B are problems in NP. We want to know if x is in A and y is in B. We'll show that A is poly-reducible to B, that is, A can be solved efficiently given access to a solver for B

18.1.1 Example 1

Reduction of Independent Set to Clique

Recall that an **independent set** is a set of vertices in G such that no two vertices are adjacent, i.e. there's no edge between them. A **clique** is a set of vertices where for each pair of vertices, there's an edge between them. A clique can be seen as the complement of an independent set.

Given graph $G = (V, E)$ and $k \in \mathbb{N}$, we can produce the complement, $\overline{G} = (V, \overline{E})$.

$$G = (V, E), \text{ there exists an independent set of size } \geq k$$
$$\iff$$
$$\overline{G} = (V, \overline{E}), \text{ there exists a clique of size } \geq k$$

Conversely, we can reduce Clique to Independent Set since the independent set is the complement of clique.

The time of reduction is $O(n^2)$, which is in polynomial time.

18.1.2 Example 2

Reduction of Vertex Cover to Independent Set

A **vertex cover** is a set of vertices such that for every edge of the graph, it is incident to at least one vertex in the set. A set of vertices is a vertex cover if and only if its complement is an independent set.

Given graph $G = (V, E)$ with n vertices and $k \in \mathbb{N}$, where there exists a vertex cover of size of at most k, we can find an independent set of size of at least $n - k$.

$$G = (V, E), \text{ there exists a vertex cover of size } \leq k$$
$$\iff$$
$$\overline{G} = (V, E), \text{ there exists an independent set of size } \geq n - k$$

Conversely, we can reduce Independent Set to Vertex Cover since the independent set is the complement of clique.

18.1.3 Example 3

Reduction of CNF SAT to Integer Linear Programming

Using the following example, we represent **True** with 1 and **False** with 0. We set each clause to be constrained to at least 1. We present $\vee$ with $+$, and the complement of x_i with $(1 - x_i)$. We constrained x_i to $[0, 1]$.

$$(x_1 \vee x_2 \vee \overline{x}_3) \ \wedge \ (x_2 \vee \overline{x}_1 \vee x_4 \vee x_5) \ \wedge \ldots$$

Integer Linear Programming

$$x_1 + x_2 + 1 - x_3 \geq 1$$
$$x_2 + (1 - x_1) + x_4 + x_5 \geq 1$$
$$\vdots$$
$$0 \leq x_i \leq 1$$

18.1.4 Example 4

Reduction of CNF SAT to CLIQUE

For every clause, set up an independent set with vertices for each x_i. Then, create an edge $(x_i, x_j) \in E$, where $x_j \neq \overline{x}_i$ and does not corresponding to the same clause.

$$(x_1 \vee x_2 \vee \overline{x}_3) \ \wedge \ (x_2 \vee x_3) \ \wedge \ (\overline{x}_1 \vee x_4 \vee x_5) \ \ldots$$

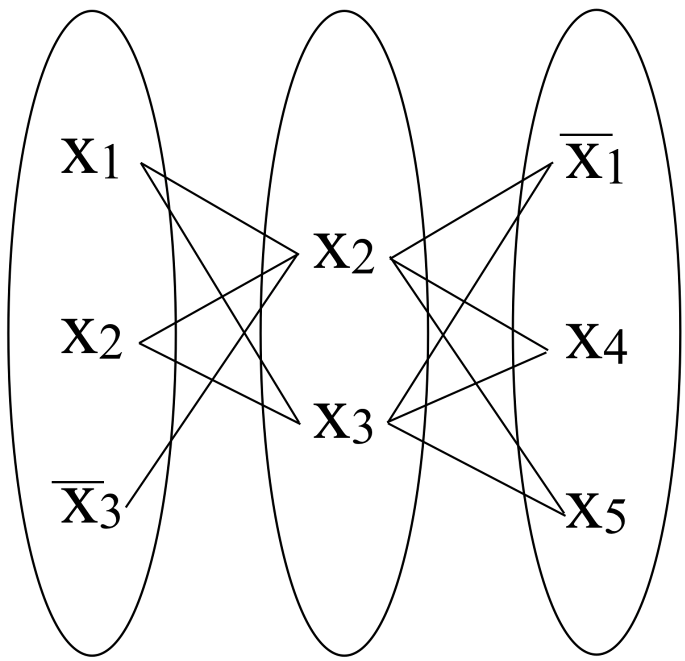

Figure 18.1: Each independent set represents a clause and each vertex represents x_i. Edge $(x_i, x_j) \in E$ if $x_j \neq \overline{x}_i$

Lemma 18.1 *F is in SAT $\iff$ G has a clique of size m*

Pick one true literal from each clause. G has an clique of size m from each clause. Set all literals in clique to ***True****, then $x_i, \overline{x_i}$ are not both in clique.*

Theorem 18.2 *If A is poly-reducible to B, and B is poly-reducible to C, then A is poly-reducible to C.*