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

Lecture 20: NP-completeness

November 13, 2019

Lecturer: Santosh Vempala *Scribe: Aditi Laddha*

20.1 Definitions

Definition 20.1 (Polynomial Time reductions) *A language A is said to polynomial time reducible to a language B or $A \leq_P B$ if there exists a polynomial time computable function $f : \Sigma^* \to \Sigma^*$ such that $\forall x \in \Sigma^*$,*

$$x \in A \iff f(x) \in B$$

Corollary 20.2 *If $A \leq_P B$ and $B \leq C$, then $A \leq_P C$.*

This is true because composition of 2 polynomials is a polynomial. So, to prove that a language, L, is NP-complete, we can either show that every language in NP reduces to L or we can show that a NP-complete language reduces to L.

Definition 20.3 (k-SUN) *k-SUN is a graph with a clique of size k(called the core) with paths of length k incident on every vertex in the core.*

$$SUN = \{(G, k) \mid G \text{ contains a } k - SUN\}$$

Definition 20.4 (Hamiltonian Cycle) *Given a graph $G = (V, E)$, a Hamiltonian cycle is a cycle in G that visits every vertex of G.*

Definition 20.5 ($DHAM$)

$$DHAM = \{G \mid G \text{ is a directed graph containing a directed Hamiltonian cycle.}\}$$

Definition 20.6 (HAM)

$$HAM = \{G \mid G \text{ is an undirected graph containing a Hamiltonian cycle.}\}$$

Definition 20.7 ($3SAT$)

$$3SAT = \{\phi \mid \phi \text{ is a boolean formula in conjunctive normal form in which all clauses have 3 literals.}\}$$

20.2 NP-complete Reductions

Theorem 20.8 *$3SAT$ is NP-complete.*

Proof: $3SAT$ is in NP. The certificate is a satisfying assignment and we can verify the satisfiability of the assignment in linear time. To prove that $3SAT$ is NP-hard, it suffices to show a recuction from SAT to $3SAT$.
Consider a SAT formula $\phi = C_1 \wedge C_2 \ldots \wedge C_m$ with $C_i = x_1 \vee x_2 \ldots \vee x_k$ where each x_j is a literal. Let $D_1 = (x_1 \vee x_2 \vee y_1) \wedge (\overline{y_1} \vee x_3 \ldots \vee x_k)$. Then

$$\exists x_1, \ldots, x_k \text{ such that } C_i = 1 \iff \exists y_1, x_1, \ldots, x_k \text{ such that } D_1 = 1$$

- ($\Rightarrow$) $\exists x_1, \ldots, x_k$ such that $C_i = 1$, then at least one of $\{x_1, \ldots, x_k\}$ must be 1. If x_1 or $x_2 = 1$ in the satisfying assignment, setting $y_1 = 0$ will satisfy D_1. Otherwise, setting $y_1 = 1$ will satisfy D_1.
- ($\Leftarrow$) $\exists y_1, x_1, \ldots, x_k$ such that $D_1 = 1$, then if $y_1 = 1$ in this assignment, at least one of $\{x_3, \ldots, x_k\}$ must also be 1. Thus, $x_1, \ldots, x_k$ satisfies C_i. Otherwise, $y_1 = 0$ in the satisfying assignment, then at least one of $\{x_1, x_2\}$ must also be 1. Again, $x_1, \ldots, x_k$ satisfies C_i.

We can recursively break down the second clause of D_1 to get a new formula D_2 equivalent to D_1 and so on until all the clauses are of length at most 3 while maintaining satisfiability. This gives

$$D_{k-3} = (x_1 \vee x_2 \vee y_1) \wedge (\overline{y_1} \vee x_3 \ldots y_2) \wedge (\overline{y_2} \vee x_4 \vee y_3) \wedge (\overline{y_3} \vee x_5 \vee y_4) \ldots (\overline{y_{k-3}} \vee x_{k-1} \vee x_k)$$

Complexity: We introduced $k-3$ new variables for a clause of length k and $k-2$ clauses for every clause having length $k \geq 4$. Here, k is bounded by n. So, in the new formula, number of clauses is $\leq nm$ and number of variables is $\leq n + nm$. Hence, this is a polynomial reduction. ■

Theorem 20.9 *SUN is* NP*-complete.*

Proof: *SUN* $\in$ NP. Given a graph $G = (V, E)$, the certificate is a subset $X \subseteq V$ and we can verify that X is a k-*SUN* in G in polynomial time. We prove NP-hardness by reducing *CLIQUE* to *SUN*. Consider a graph $G = (V, E)$. The goal is to find whether G contains a clique of size at least k. We construct a new graph $G' = (V', E')$ with $V \subseteq V', E \subseteq E'$ and V' contains kn additional vertices in forms of disjoint k-paths starting from every vertex in V. Then,

$$(G, k) \in CLIQUE \Longleftrightarrow (G', k) \in SUN$$

- ($\Rightarrow$) G contains a clique $X \subseteq V$ of size k, then $X \cup \{$ k-paths in $V' \backslash V$ having a vertex in X as one of its end-points $\}$ forms a k-SUN in G'.
- ($\Leftarrow$) G' contains a k-SUN, X'. The core of X' must belong to V as every vertex in the core has degree $k-1$ but vertices in $V' \backslash V$ have degrees 1 or 2. So, the core of X' forms a k-clique in G.

■

Theorem 20.10 *HAM* $\leq_P$ *DHAM.*

Proof: Consider an undirected graph $G = (V, E)$ with $|V| = n, |E| = n$. We construct a directed graph $G' = (V, E')$ such that $E' = \{(u \to v), (v \to u) : (u, v) \in E\}$.

$$G \in HAM \Longleftrightarrow G' \in DHAM$$

Consider a Hamiltonian cycle

$$C = v_1 - v_2 - \ldots - v_n - v_1$$

in G. Then

$$C' = v_1 \to v_2 \to \ldots \to v_n \to v_1$$

is a directed Hamiltonian Cycle in G' and vice versa. Also, $|V'| = n$ and $|E'| = 2m$. So, size of G' is polynomial in size of G and hence this is a polynomial reduction. ■

Theorem 20.11 *and DHAM* $\leq_P$ *HAM.*

Proof: Consider an directed graph $G = (V, E)$ with $|V| = n, |E| = n$. We construct a directed graph $G' = (V', E')$ such that for every $v \in V$, there are three vertices $v_i, \hat{v}, v_o \in V'$ and $(v_i, \hat{v}), (\hat{v}, v_o) \in E'$. For every edge $u \to v \in E$, we add $(u_o, v_i) \in E'$. Then,

$$G \in DHAM \Longleftrightarrow G' \in HAM$$

Consider a Hamiltonian cycle

$$C = v_1 \to v_2 \to \ldots \to v_n \to v_1$$

in G. Then

$$C' = v_{1i} - \hat{v}_{1i} - v_{1o} - v_{2i} - \hat{v}_{2i} - v_{2o} - \ldots - v_{ni} - \hat{v}_{ni} - v_{no} - v_{1i}$$

is a Hamiltonian Cycle in G'.

Consider a Hamiltonian Cycle, C' in G'. In this cycle every vertex has degree 2 and hence every vertex of the form $\hat{v}$ must be adjacent to v_i and v_o as they are its only neighbors. So every vertex of the type v_o is adjacent to exactly one vertex of the type w_i in this cycle. This implies that $v \to w$ is an edge in G. And every vertex of the type v_i is adjacent to exactly one vertex of the type w_o in this cycle. This implies that $w \to v$ is an edge in G. Thus C' must be of the form

$$C' = v_{1i} - \hat{v}_{1i} - v_{1o} - v_{2i} - \hat{v}_{2i} - v_{2o} - \ldots - v_{ni} - \hat{v}_{ni} - v_{no} - v_{1i}$$

and hence

$$C = v_1 \to v_2 \to \ldots \to v_n \to v_1$$

is a Hamiltonian cycle in G. Also, $|V'| = 3n$ and $|E'| = m + 2n$. So, size of G' is polynomial in size of G and hence this is a polytime reduction. ■

Theorem 20.12 *$DHAM$ is* NP*-complete.*

$DHAM$ is in NP. Given a graph $G = (V, E)$, the certificate is a Hamiltonian cycle in G and we can verify that it is a Hamiltonian cycle in polynomial time. Now, we prove that $DHAM$ is NP-hard by reducing SAT to $DHAM$.

Given a formula ϕ with variables $X = \{x_1, \ldots, x_n\}$ and clauses $C = \{C_1, \ldots, C_m\}$, we create a graph $G(\phi)$ by creating "gadgets" for every variable and clause of ϕ.

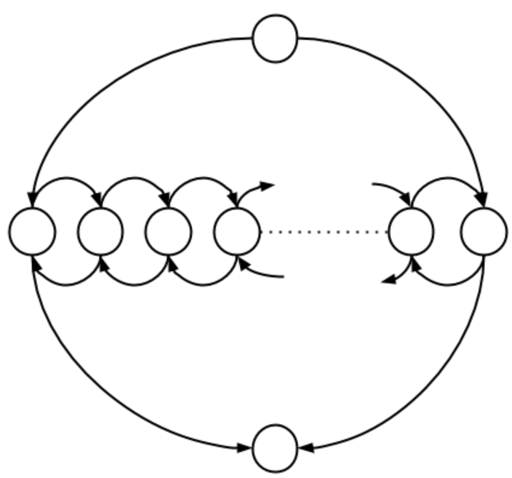

Figure 20.1: Gadget for a variable x_i.

For variable $x_i \in X$, let $N_i = |\{C_j : x_i \in C_j \vee \overline{x_i} \in C_j\}|$. The gadget for x_i is a graph, G_i consisting of a bi-directed path with $2N_i$ nodes. We add 2 more nodes to this gadget, one having edges to the end nodes of the bi-directed path and the other having edges from them. We then connect the gadget graphs G_i corresponding to x_i in a directed cycle $G_1 \to G_2 \to \ldots G_n \to G_1$. The idea is that if x_i is *true* in an assignment for ϕ then we can traverse the gadget G_i from left to right, and right to left if x_i is *false*. We can decide the direction for traversing each gadget independently depending on the value of the corresponding variable.

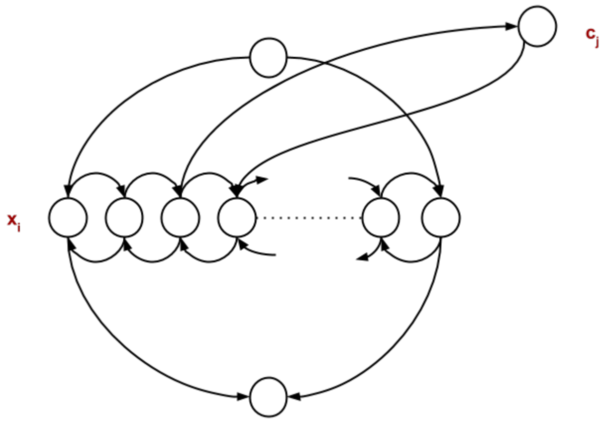

Figure 20.2: C_j contains x_i.

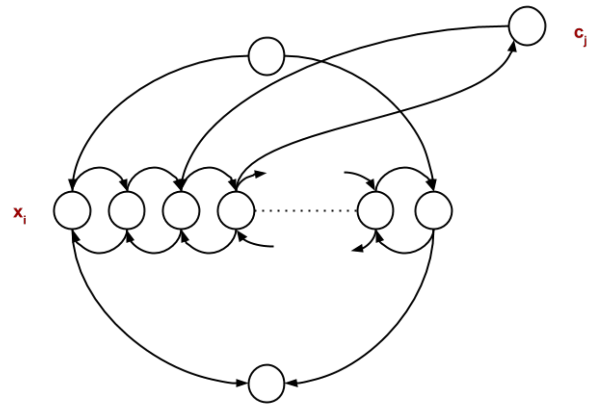

Figure 20.3: C_j contains $\overline{x_i}$.

The gadget for a clause $C_j \in C$ consists of a node c_j. For every variable x_i which appears as a positive literal in C_j, we add edges from the jth pair of nodes in the vertex gadget G_i to c_j in a clockwise direction and for every variable x_i which appears as a negative literal, $\overline{x_i}$ in C_j, we add edges from the jth pair of nodes in the vertex gadget G_i to c_j in a counter clockwise direction.

$$\phi \in SAT \iff G(\phi) \in DHAM$$

If ϕ has a satisfying assignment, A then for each $i \in [n]$ we traverse G_i from left to right if x_i is $true$ in the A and right to left if x_i is $false$ in A. To move from one gadget to the next, we use the cycle $G_1 \rightarrow G_2 \Rightarrow \ldots G_n \rightarrow G_1$. This gives a directed cycle covering all vertices except c_j's. Since A satisfies ϕ, we can assign a true literal for every clause C_j. If we assigned x_i to C_j, then we can detour at the jth pair in the ith gadget to visit the node c_j. As x_i is true, the cycle visits G_i left to right and our construction ensures that we can visit c_j in the correct order. Similarly, if we assigned $\overline{x_i}$ to C_j, then we can again detour at the jth pair in the ith gadget to visit the node c_j. As x_i must be false, the cycle visits G_i right to left and our construction again ensures that we can visit c_j in the correct order.
For details about the proof in other direction, please see reference.
Complexity: For every variable x_i, G contains $2N_i + 2 \leq 2m + 2$ vertices and G contains one vertex for each clause. So, total number of vertices in G is $\leq n(2m + 2) + m = O(nm)$. Every gadget G_i contains $O(m)$ edges and a clause node has at most $2n$ edges incident to it. There are also n edges connecting the gadget graphs to each other. So, total number of edges in G is $\leq O(mn)$. Hence, this is a polynomial time reduction.

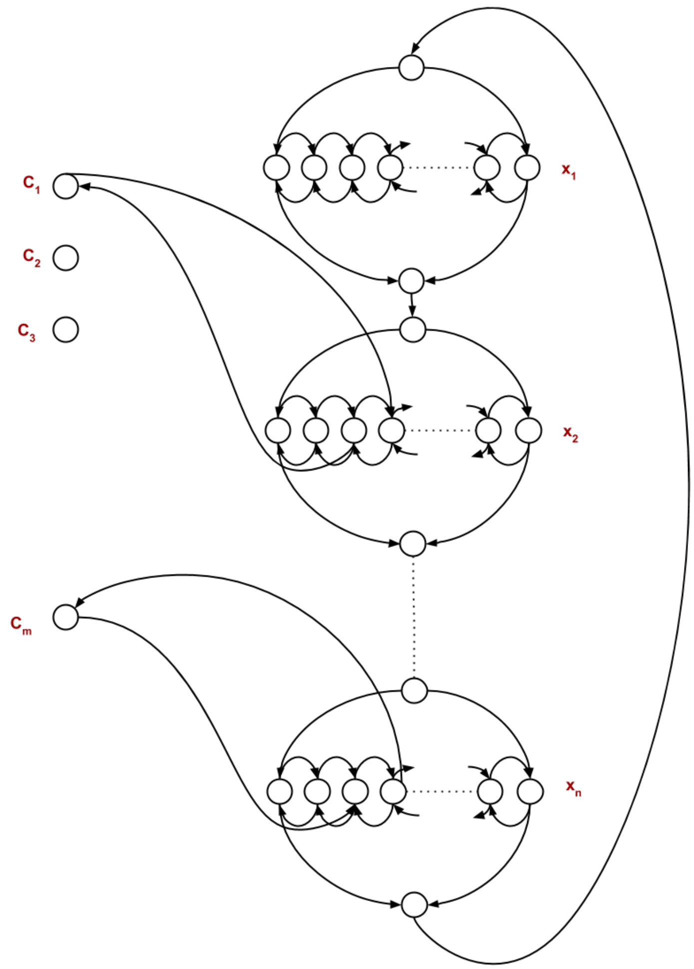

Figure 20.4: The graph $G(\phi)$.

20.3 References

- Ch 7.5 Additional NP-Complete Problems, "Introduction to the Theory of Computation"