# Chapter 23

# More NP-Complete Problems

**CS 473: Fundamental Algorithms, Spring 2011**
April 21, 2011

#### 23.0.0.1 Recap

NP: languages that have polynomial time certifiers/verifiers

A language $L$ is NP-Complete iff

- $L$ is in NP
- for every $L'$ in NP, $L' \leq_P L$

$L$ is NP-Hard if for every $L'$ in NP, $L' \leq_P L$.

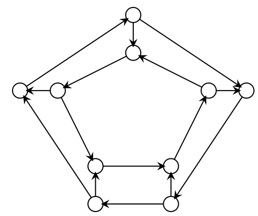
**Theorem 23.0.1 (Cook-Levin)** *Circuit-SAT and SAT are NP-Complete.*

#### 23.0.0.2 Recap contd

**Theorem 23.0.2 (Cook-Levin)** *Circuit-SAT and SAT are NP-Complete.*

Establish NP-Completeness via reductions:

- SAT $\leq_P$ 3-SAT and hence 3-SAT is NP-complete
- 3-SAT $\leq_P$ Independent Set (which is in NP) and hence Independent Set is NP-complete
- Vertex Cover is NP-complete
- Clique is NP-complete
- Set Cover is NP-Complete



#### 23.0.0.3 Today

Prove

- Hamiltonian Cycle Problem is NP-Complete
- 3-Coloring is NP-Complete

#### 23.0.0.4 Directed Hamiltonian Cycle

**Input** Given a directed graph $G = (V, E)$ with $n$ vertices

**Goal** Does $G$ have a *Hamiltonian cycle*?

- A Hamiltonian cycle is a cycle in the graph that visits every vertex in $G$ exactly once

#### 23.0.0.5 Directed Hamiltonian Cycle is $NP$-complete

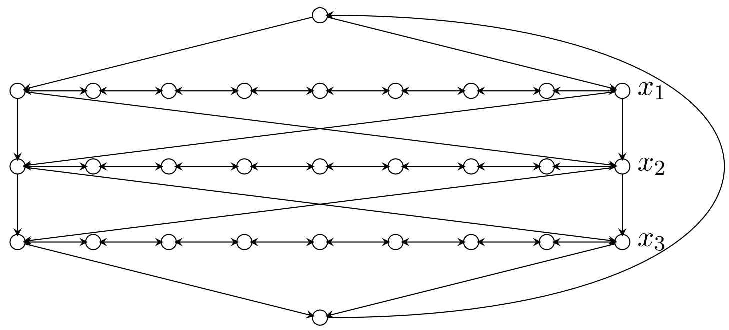
- Directed Hamiltonian Cycle is in $NP$
  - *Certificate:* Sequence of vertices
  - *Certifier:* Check if every vertex (except the first) appears exactly once, and that consecutive vertices are connected by a directed edge
- *Hardness:* We will show 3-SAT $\leq_P$ Directed Hamiltonian Cycle

#### 23.0.0.6 Reduction

Given 3-SAT formula $\varphi$ create a graph $G_\varphi$ such that

- $G_\varphi$ has a Hamiltonian cycle if and only if $\varphi$ is satisfiable
- $G_\varphi$ should be constructible from $\varphi$ by a polynomial time algorithm $\mathcal{A}$

*Notation:* $\varphi$ has $n$ variables $x_1, x_2, \ldots, x_n$ and $m$ clauses $C_1, C_2, \ldots, C_m$.




#### 23.0.0.7 Reduction: First Ideas

- Viewing SAT: Assign values to $n$ variables, and each clauses has 3 ways in which it can be satisfied
- Construct graph with $2^n$ Hamiltonian cycles, where each cycle corresponds to some boolean assignment
- Then add more graph structure to encode constraints on assignments imposed by the clauses

#### 23.0.0.8 The Reduction: Phase I

- Traverse path $i$ from left to right iff $x_i$ is set to true
- Each path has $3(m+1)$ nodes where $m$ is number of clauses in $\varphi$; nodes numbered from left to right (1 to $3m+3$)

#### 23.0.0.9 The Reduction: Phase II

- Add vertex $c_j$ for clause $C_j$. $c_j$ has edge *from* vertex $3j$ and *to* vertex $3j+1$ on path $i$ if $x_i$ appears in clause $C_j$, and has edge *from* vertex $3j+1$ and *to* vertex $3j$ if $\neg x_i$ appears in $C_j$.

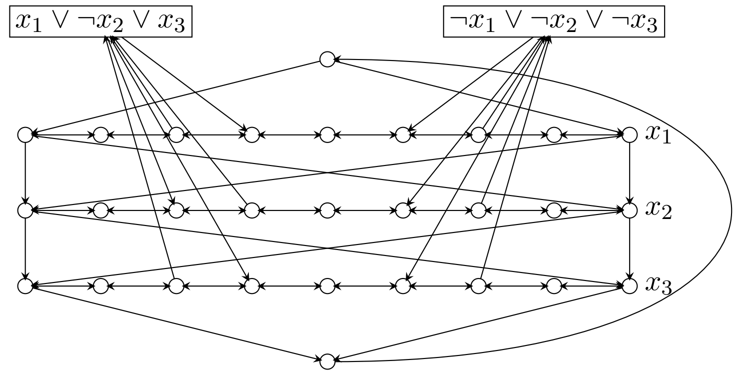
#### 23.0.0.10 Correctness Proof

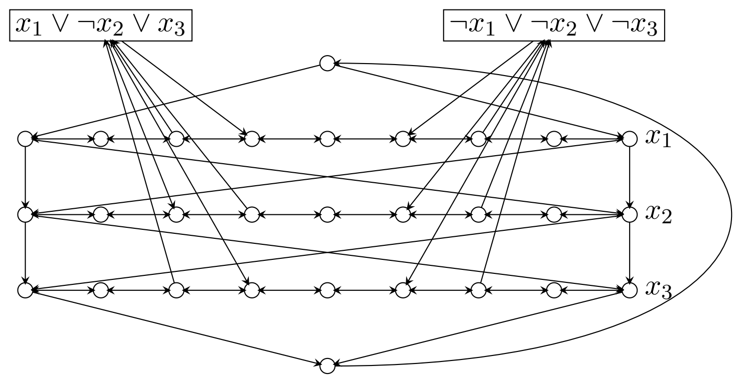
**Proposition 23.0.3** *$\varphi$ has a satisfying assignment iff $G_\varphi$ has a Hamiltonian cycle*

*Proof*:

$\Rightarrow$ Let $a$ be the satisfying assignment for $\varphi$. Define Hamiltonian cycle as follows

- If $a(x_i) = 1$ then traverse path $i$ from left to right
- If $a(x_i) = 0$ then traverse path $i$ from right to left




  - For each clause, path of at least one variable is in the "right" direction to splice in the node corresponding to clause

∎

### 23.0.0.11 Hamiltonian Cycle ⇒ Satisfying assignment

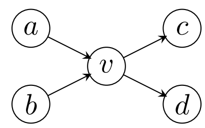
Suppose $\Pi$ is a Hamiltonian cycle in $G_\varphi$

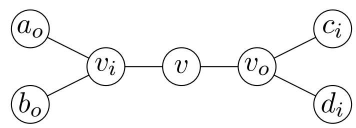
- If $\Pi$ enters $c_j$ (vertex for clause $C_j$) from vertex $3j$ on path $i$ then it must leave the clause vertex on edge to $3j+1$ on the *same path* $i$
  - If not, then only unvisited neightbor of $3j+1$ on path $i$ is $3j+2$
  - Thus, we don't have two unvisited neighbors (one to enter from, and the other to leave) to have a Hamiltonian Cycle
- Similarly, if $\Pi$ enters $c_j$ from vertex $3j+1$ on path $i$ then it must leave the clause vertex $c_j$ on edge to $3j$ on path $i$

### 23.0.0.12 Example

### 23.0.0.13 Hamiltonian Cycle ⇒ Satisfying assignment (contd)

- Thus, vertices visited immediately before and after $C_i$ are connected by an edge




- We can remove $c_j$ from cycle, and get Hamiltonian cyle in $G - c_j$
- Consider hamiltonian cycle in $G - \{c_1, \dots c_m\}$; it traverses each path in only one direction, which determines the truth assignment

#### 23.0.0.14 Hamiltonian Cycle

**Problem 23.0.4 Input** *Given undirected graph $G = (V, E)$*

**Goal** *Does G have a Hamiltonian cycle? That is, is there a cycle that visits every vertex exactly one (except start and end vertex)?*

#### 23.0.0.15 $NP$-completeness

**Theorem 23.0.5** *Hamiltonian cycle problem for undirected graphs is $NP$-complete*

*Proof*:

- The problem is in $NP$; proof left as exercise
- Hardness proved by reducing Directed Hamiltonian Cycle to this problem

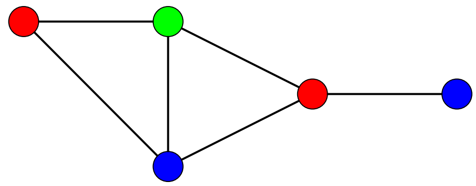
■

#### 23.0.0.16 Reduction Sketch

*Goal:* Given directed graph $G$, need to construct undirected graph $G'$ such that $G$ has Hamiltonian Path iff $G'$ has Hamiltonian path

**Reduction**

- Replace each vertex $v$ by 3 vertices: $v_{in}, v$, and $v_{out}$
- A directed edge $(a, b)$ is replaced by edge $(a_{out}, b_{in})$



#### 23.0.0.17 Reduction: Wrapup

- The reduction is polynomial time (exercise)
- The reduction is correct (exercise)

#### 23.0.0.18 Graph Coloring

**Input** Given an undirected graph $G = (V, E)$ and integer $k$

**Goal** Can the vertices of the graph be colored using $k$ colors so that vertices connected by an edge do not get the same color?

#### 23.0.0.19 Graph 3-Coloring

**Input** Given an undirected graph $G = (V, E)$

**Goal** Can the vertices of the graph be colored using 3 colors so that vertices connected by an edge do not get the same color?

#### 23.0.0.20 Graph Coloring

*Observation:* If $G$ is colored with $k$ colors then each color class (nodes of same color) form an independent set in $G$. Thus, $G$ can be partitioned into $k$ independent sets iff $G$ is $k$-colorable.

Graph 2-Coloring can be decided in polynomial time.

$G$ is 2-colorable iff $G$ is bipartite! There is a linear time algorithm to check if $G$ is bipartite using BFS (see Chapter 3 of Kleiberg-Tardos book).

### 23.0.0.21 Graph Coloring and Register Allocation

**Register Allocation**
Assign variables to (at most) $k$ registers such that variables needed at the same time are not assigned to the same register

**Interference Graph**
Vertices are variables, and there is an edge between two vertices, if the two variables are "live" at the same time.

**Observations**

- *[Chaitin]* Register allocation problem is equivalent to coloring the interference graph with $k$ colors
- Moreover, 3-COLOR $\leq_P$ k-Register Allocation, for any $k \geq 3$

### 23.0.0.22 Class Room Scheduling

Given $n$ classes and their meeting times, are $k$ rooms sufficient?
Reduce to Graph $k$-Coloring problem
Create graph $G$

- a node $v_i$ for each class $i$
- an edge between $v_i$ and $v_j$ if classes $i$ and $j$ *conflict*

Exercise: $G$ is $k$-colorable iff $k$ rooms are sufficient

### 23.0.0.23 Frequency Assignments in Cellular Networks

Cellular telephone systems that use Frequency Division Multiple Access (FDMA) (example: GSM in Europe and Asia and AT&T in USA)

- Breakup a frequency range $[a, b]$ into disjoint *bands* of frequencies $[a_0, b_0], [a_1, b_1], \ldots, [a_k, b_k]$
- Each cell phone tower (simplifying) gets one band
- Constraint: nearby towers cannot be assigned same band, otherwise signals will interference

*Problem:* given $k$ bands and some region with $n$ towers, is there a way to assign the bands to avoid interference?

Can reduce to $k$-coloring by creating intereference/conflict graph on towers

### 23.0.0.24 3-Coloring is NP-Complete

- 3-Coloring is in $NP$
  - *Certificate:* for each node a color from $\{1, 2, 3\}$
  - *Certifier:* Check if for each edge $(u, v)$, the color of $u$ is different from that of $v$
- *Hardness:* We will show 3-SAT $\leq_P$ 3-COLORING

### 23.0.0.25 Reduction Idea

Start with 3-SAT formula $\varphi$ with $n$ variables $x_1, \ldots, x_n$ and $m$ clauses $C_1, \ldots, C_m$. Create graph $G_\varphi$ such that $G_\varphi$ is 3-colorable iff $\varphi$ is satisfiable

¡+-¿ need to establish truth assignment for $x_1, \ldots, x_n$ via colors for some nodes in $G_\varphi$.

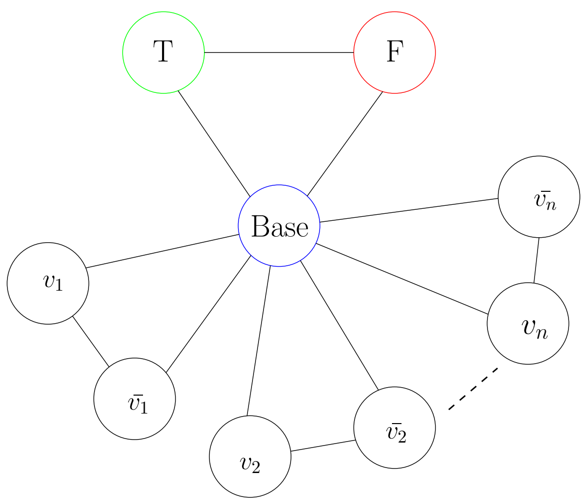
¡+-¿ create triangle with node True, False, Base

¡+-¿ for each variable $x_i$ two nodes $v_i$ and $\bar{v_i}$ connected in a triangle with common Base

¡+-¿ If graph is 3-colored, either $v_i$ or $\bar{v_i}$ gets the same color as True. Interpret this as a truth assignment to $v_i$

¡+-¿ Need to add constraints to ensure clauses are satisfied (next phase)

### 23.0.0.26 Figure



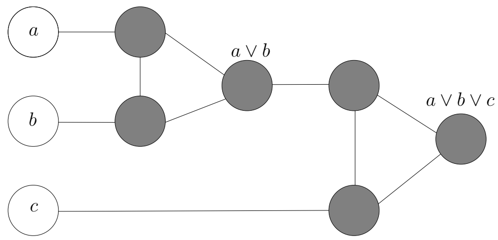

### 23.0.0.27 Clause Satisfiability Gadget

For each cluase $C_j = (a \vee b \vee c)$, create a small gadget graph

- gadget graph connects to nodes corresponding to $a, b, c$

- needs to implement OR

OR-gadget-graph:



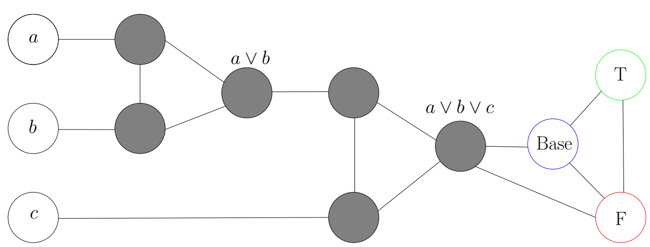

### 23.0.0.28 OR-Gadget Graph

*Property:* if $a, b, c$ are colored False in a 3-coloring then output node of OR-gadget has to be colored False.

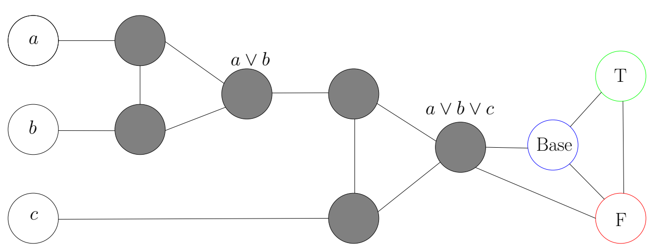
*Property:* if one of $a, b, c$ is colored True then OR-gadget can be 3-colored such that output node of OR-gadget is colored True.

### 23.0.0.29 Reduction

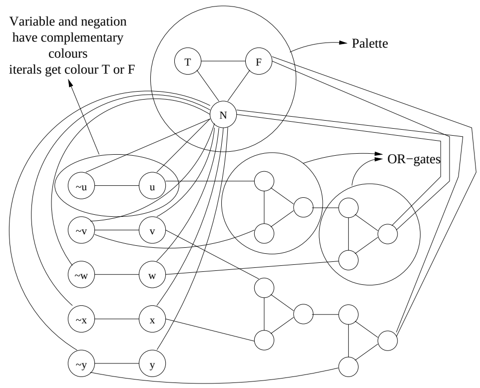
- create triangle with nodes True, False, Base
- for each variable $x_i$ two nodes $v_i$ and $\bar{v}_i$ connected in a triangle with common Base
- for each clause $C_j = (a \vee b \vee c)$, add OR-gadget graph with input nodes $a, b, c$ and connect output node of gadget to both False and Base




### 23.0.0.30 Reduction




**Claim 23.0.6** *No legal 3-coloring of above graph (with coloring of nodes $T, F, B$ fixed) in which $a, b, c$ are colored False. If any of $a, b, c$ are colored True then there is a legal 3-coloring of above graph.*




### 23.0.0.31 Reduction Outline

**Example 23.0.7** $\varphi = (u \vee \neg v \vee w) \wedge (v \vee x \vee \neg y)$

### 23.0.0.32 Correctness of Reduction

$\varphi$ is satisfiable implies $G_\varphi$ is 3-colorable

¡+-¿ if $x_i$ is assigned True, color $v_i$ True and $\bar{v}_i$ False

¡+-¿ for each clause $C_j = (a \vee b \vee c)$ at least one of $a, b, c$ is colored True. OR-gadget for $C_j$ can be 3-colored such that output is True.

$G_\varphi$ is 3-colorable implies $\varphi$ is satisfiable

¡+-¿ if $v_i$ is colored True then set $x_i$ to be True, this is a legal truth assignment

¡+-¿ consider any clause $C_j = (a \vee b \vee c)$. it cannot be that all $a, b, c$ are False. If so, output of OR-gadget for $C_j$ has to be colored False but output is connected to Base and False!

### 23.0.0.33 Other NP-Complete Problems

- 3-Dimensional Matching
- Subset Sum

Read book.

### 23.0.0.34 Need to Know NP-Complete Problems

- 3-SAT
- Circuit-SAT
- Independent Set