

# Theory of Computation

## Chapter 9

Guan-Shieng Huang

May 12, 2003

# NP-completeness Problems

**NP:** the class of languages decided by nondeterministic Turing machine in polynomial time

**NP-completeness:**

Cook's theorem: SAT is NP-complete.

**Certificate of TM:**

Hard to find an answer if there is one, but easy to verify.

SAT — a satisfying truth assignment

HAMILTON PATH — a Hamilton path

## Variants of Satisfiability

- $k$ -SAT
- 3-SAT
- 2-SAT
- MAX 2SAT
- NAESAT

**$k$ -SAT:** Each clause has at most  $k$  literals.

$$(l_1 \vee l_2 \vee \cdots \vee l_t, t \leq k)$$

**Proposition 9.2** 3-SAT is NP-complete.

For any clause  $C = l_1 \vee l_2 \vee \cdots \vee l_t$ , we introduce a **new variable**  $x$  and split  $C$  into

$$C_1 = l_1 \vee l_2 \vee \cdots \vee l_{t-2} \vee x,$$

$$C_2 = \neg x \vee l_{t-1} \vee l_t.$$

Each time we obtain a clause with 3 literals. Then  $F \wedge C$  is satisfiable iff  $F \wedge C_1 \wedge C_2$  is satisfiable

**Proposition 9.3** 3-SAT remains NP-complete if each variable is restricted to appear at most three times, and each literal at most twice.

Suppose a variable  $x$  appears  $k$  times. Replace the  $i$ th  $x$  by new variable  $x_i$  for  $1 \leq i \leq k$ , and add

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge \cdots \wedge (\neg x_k \vee x_1)$$

to the expression.

$$(x_1 \Rightarrow x_2) \wedge (x_2 \Rightarrow x_3) \wedge \cdots \wedge (x_k \Rightarrow x_1)$$

$\therefore x_i$  equals  $x_j$  for  $1 \leq i, j \leq k$ .

**Theorem** 2-SAT is in NL.

**Corollary** 2-SAT is in P.

**MAX 2SAT:** Find a truth assignment that satisfies the most clauses where each clause contains at most two literals.

**Theorem 9.2** MAX 2SAT is NP-complete.

Reduce 3-SAT to MAX 2SAT.

For any clause  $x \vee y \vee z$  where  $x, y, z$  are literals, translate it into

$$\begin{aligned} & x, y, z, w, \\ & \neg x \vee \neg y, \neg y \vee \neg z, \neg z \vee \neg x, \\ & x \vee \neg w, y \vee \neg w, z \vee \neg w. \end{aligned}$$

Then  $x \vee y \vee z$  is satisfied iff 7 clauses are satisfied.

Let  $F$  be an instance of 3-SAT with  $m$  clauses. Then  $F$  is satisfiable iff  $7m$  clauses can be satisfied in  $R(F)$ .



**NAESAT:** A clause is satisfied iff not all literals are true, and not all false. (Eg,  $x \vee \neg y \vee z$ , not  $\{x=1, y=0, z=1\}$   $\{x=0, y=1, z=0\}$ )

**Theorem 9.3** NAESAT is NP-complete.

The reduction from CIRCUIT SAT to SAT is indeed a proof.

**Independent set (in a graph):**

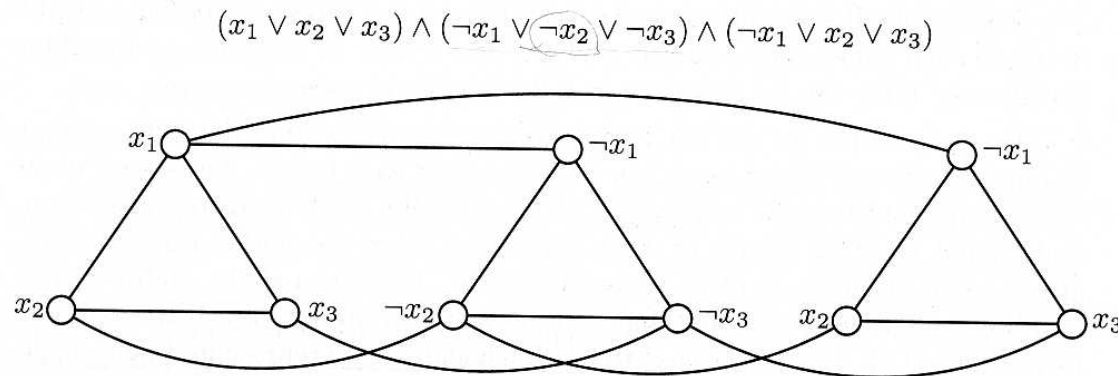
$G = (V, E), I \subseteq V$ .  $I$  is an independent set of  $G$  iff for all  $i, j \in I, (i, j) \notin E$ .

**INDEPENDENT SET:** Given a graph  $G$  and a number  $k$ , is there an independent set  $I$  of  $G$  with  $|I| \geq k$ ?

**Theorem 9.4** INDEPENDENT SET is NP-complete.

Reduce 3-SAT to it. If there are  $m$  clauses, let  $k = m$ .

1. Each clause corresponds to one triangle.
2. Complement literals are joined by an arc.



**Figure 9-2.** Reduction to INDEPENDENT SET.

**Corollary** 4-DEGREE INDEPENDENT SET is NP-complete.

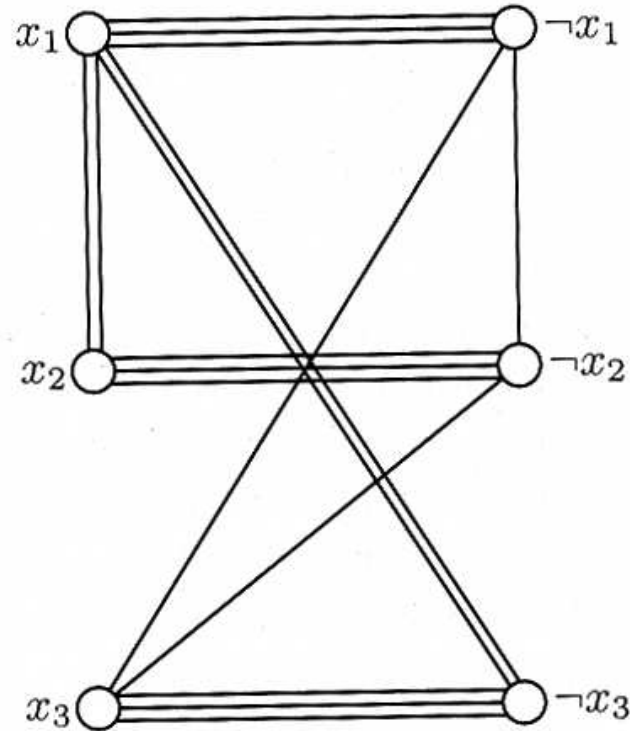
**Clique:**  $G = (V, E)$ ,  $C \subseteq V$ .  $C$  is a **clique** of  $G$  iff for all  $i, j \in C$ ,  $(i, j) \in E$ .

**Corollary** CLIQUE is NP-complete.

**Node Cover:**  $G = (V, E)$ ,  $N \subseteq V$  is a **node cover** iff for every edge  $(i, j) \in E$ , either  $i \in N$  or  $j \in N$ .

**Corollary** NODE COVER is NP-complete.

**Cut:**  $G = (V, E)$ ,  $S \subseteq V$ , then  $(S, V - S)$  is a **cut**. The size of a cut is the number of edges between  $S$  and  $V - S$ .



**Figure 9-3.** Reduction to MAX CUT.

**Theorem 9.5** MAX CUT is NP-complete. Reduce NAESAT to it.

1.  $F = \{C_1, C_2, \dots, C_m\}$  clauses, each contains three literals. The variables are  $x_1, x_2, \dots, x_n$ .  
 $\Rightarrow G$  has  $2n$  nodes, namely,  $x_1, \dots, x_n, \neg x_1, \dots, \neg x_n$ .
2. (a) For a clause  $C_i = \alpha \vee \beta \vee \gamma$ , add edges  $(\alpha, \beta)$ ,  $(\alpha, \gamma)$ ,  $(\beta, \gamma)$  into  $G$ . For a clause  $C_i = \alpha \vee \alpha \vee \beta$ , add  $(\alpha, \beta)$ ,  $(\alpha, \beta)$  into  $G$ .  
(b) For any variable  $x_i$ , let  $n_i$  be the number of occurrence of  $x_i$  and  $\neg x_i$ . Add  $n_i$  edges between  $x_i$  and  $\neg x_i$ . (**3m edges are added in total.**)
3. If  $F$  is NAESAT, let  $S$  be the set of literals that is true.

Then  $(S, V - S)$  is a cut of size

$$2m + 3m = 5m.$$

4. If  $G$  has a cut  $S$  of size  $5m$  or more, without loss of generality, we assume  $x_i$  and  $\neg x_i$  are in different side. There are exactly  $3m$  edges introduced in 2.(b). There are at most  $2m$  edges introduced in 2.(a), which equals to  $2m$  if and only if all clauses are NAESAT.



**Max Bisection:** A special MAX CUT with  $|S| = |V - S|$ .

**Lemma 9.1** MAX BISECTION is NP-complete.

Indeed, the proof of Theorem 9.5 is a one. Or, simply add  $|V|$  isolated nodes into  $G$ .

**Bisection Width:** Separate the nodes into two equal parts with minimum cut.

**Remark** It is a generalization of MIN CUT, which is in P.  
(MAX FLOW=MIN CUT).

**Theorem 9.6** BISECTION WIDTH is NP-complete.

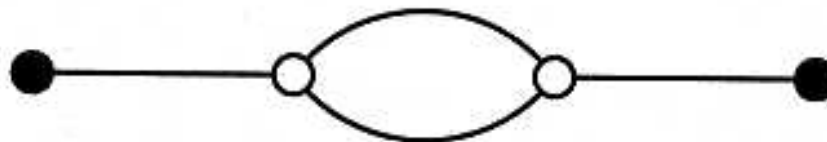
Let  $G = (V, E)$  where  $|V| = 2n$ , then  $G$  has a bisection of size  $k$  if and only if the complement of  $G$  has a bisection of size  $n^2 - k$ .

**Hamilton Path:** Given an undirected graph  $G$ , does it have a Hamilton path?

**Theorem** HAMILTON PATH is NP-complete.

Reduce 3-SAT to it.

1. choice gadget



2. consistency gadget

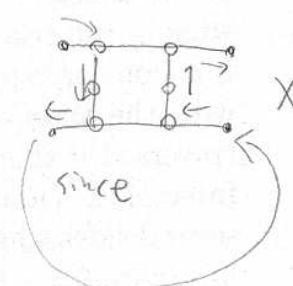
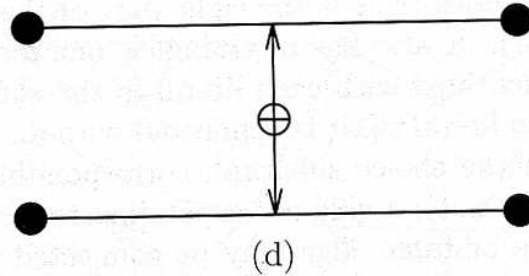
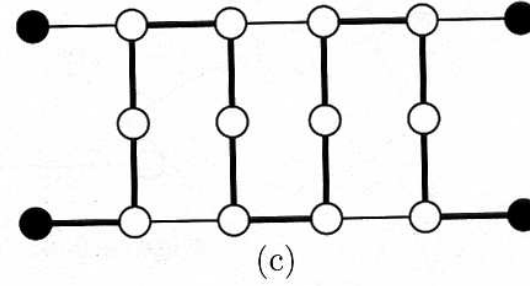
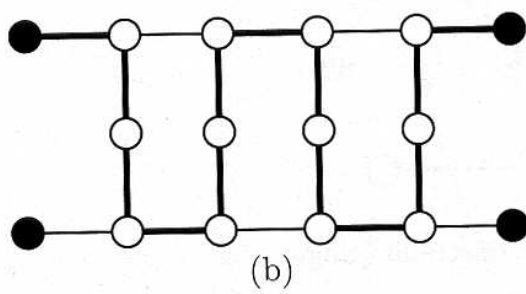
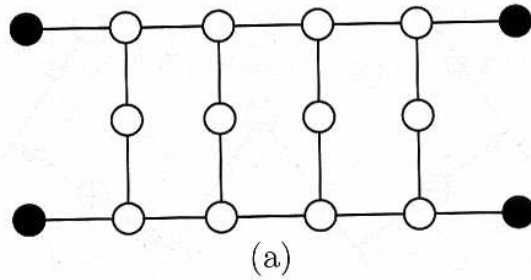


Figure 9-5. The consistency gadget.

3. constraint gadget

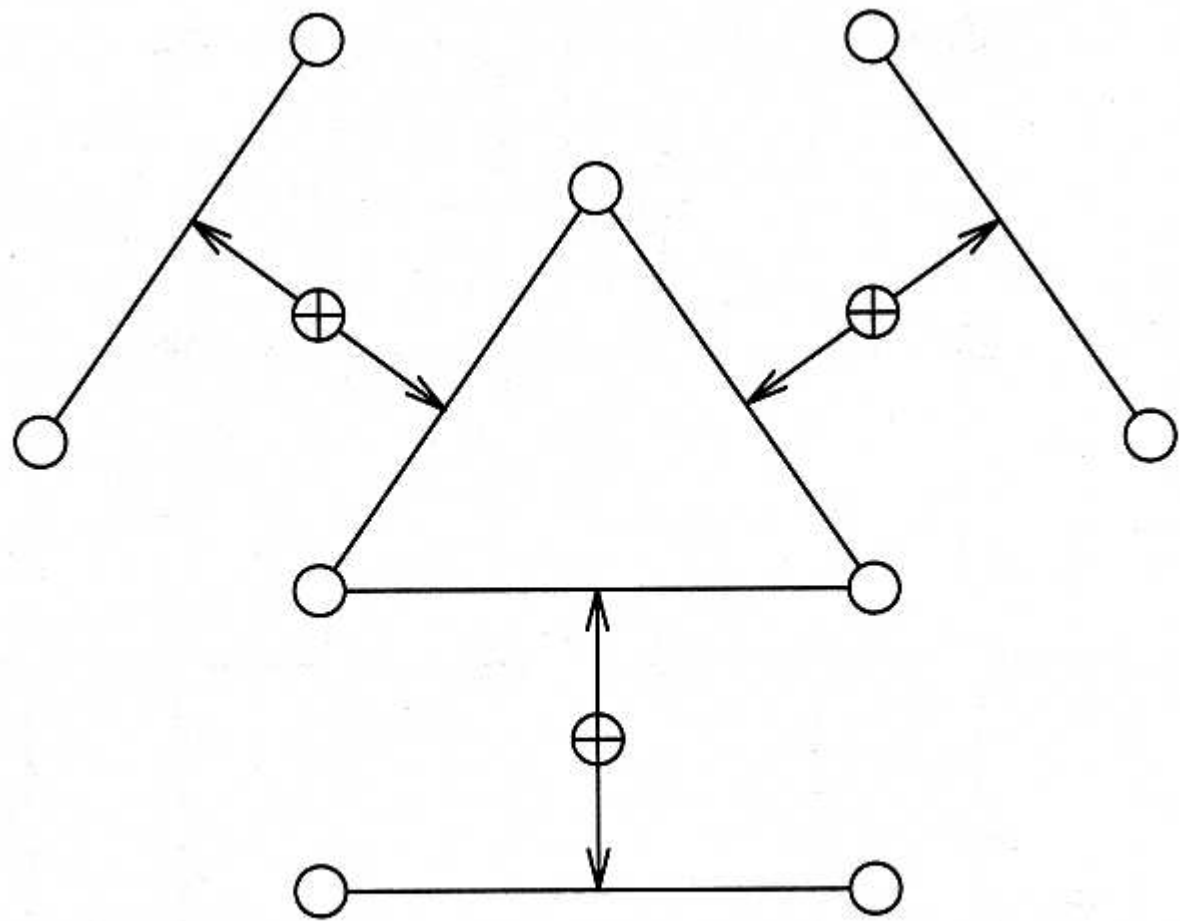


Figure 9-6. The constraint gadget.

#### 4. Reduction from 3-SAT to HAMILTON PATH:

- (a) Start from node 1, end with node 2.
- (b) All  $\odot$  nodes are connected in a big clique.

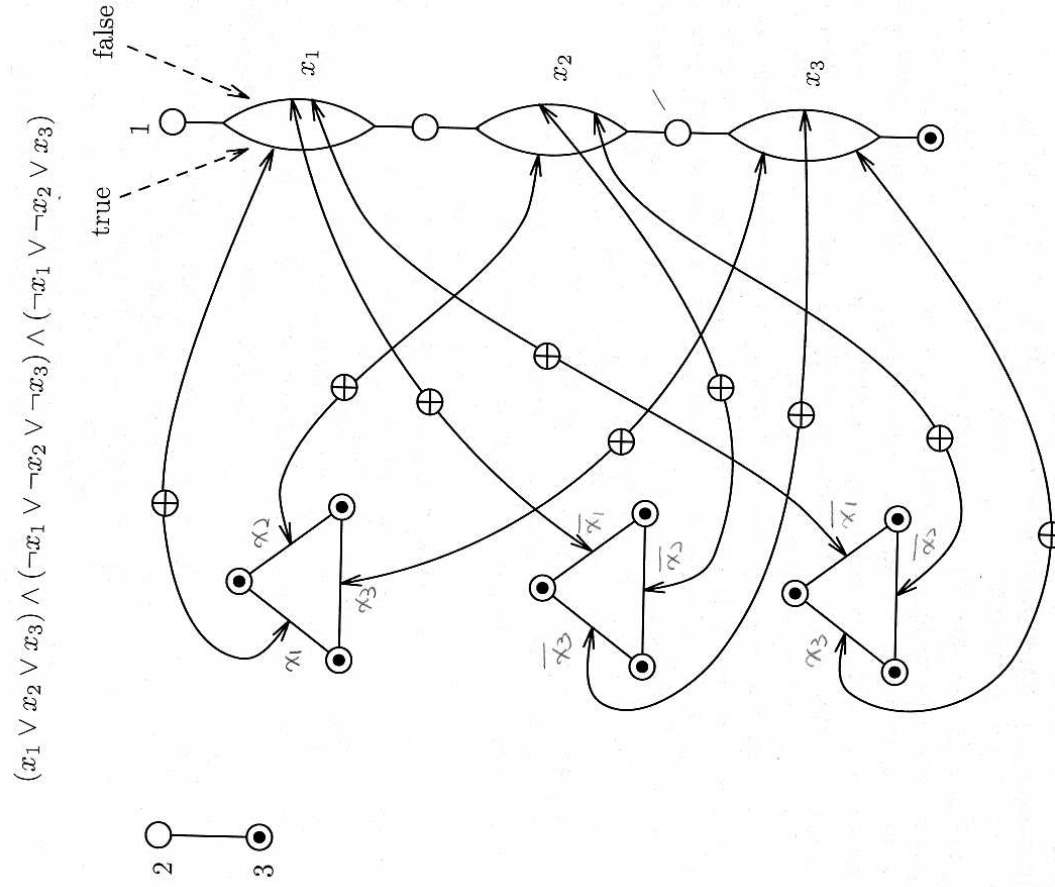


Figure 9-7. The reduction from 3SAT to HAMILTON PATH.

**Corollary** TSP(D) is NP-complete.

Reduce HAMILTON PATH to it.

$$d(i, j) = \begin{cases} 1 & \text{if } (i, j) \text{ is an edge in } G; \\ 2 & \text{otherwise.} \end{cases}$$

We also add an extra node that connects to other nodes with distance 1.

**$k$ -coloring of a graph:** Color a graph with at most  $k$  colors such that no two adjacent nodes have the same color.

**Theorem 9.8** 3-COLORING is NP-complete.



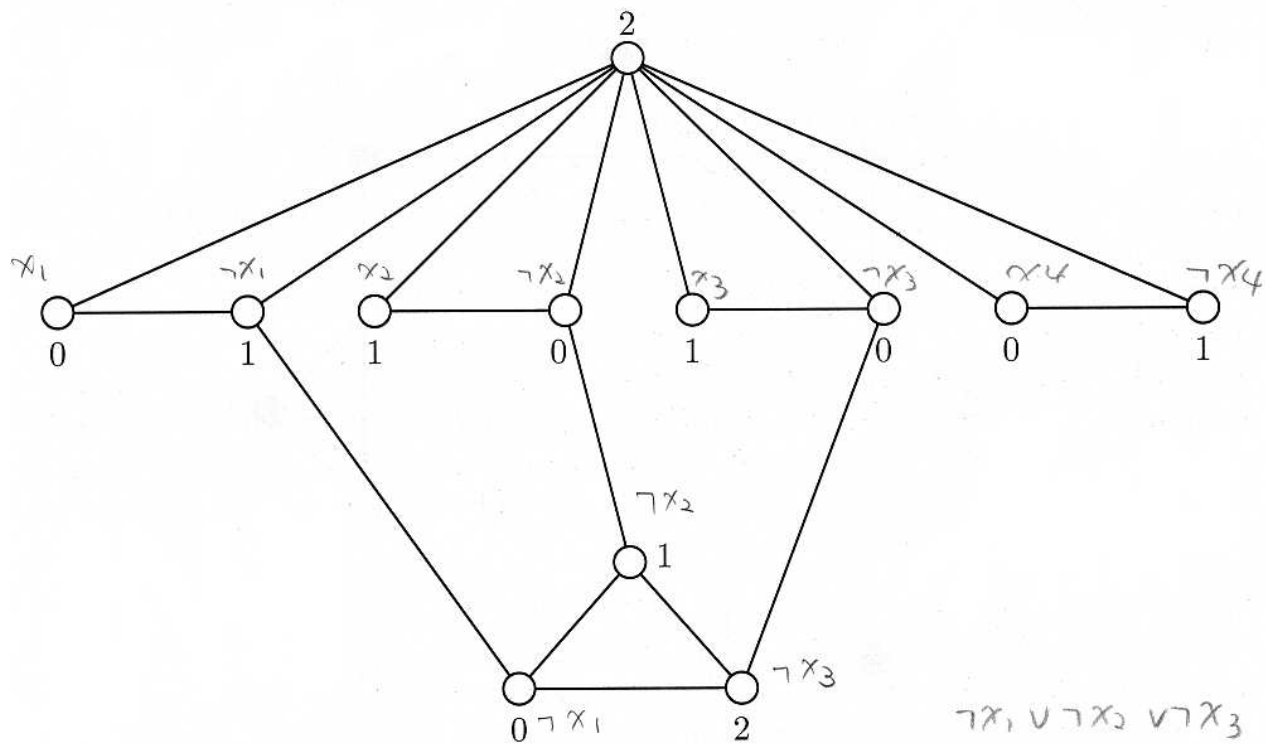


Figure 9-8. The reduction to 3-COLORING.

Reduce NAESAT to it.

1. choice gadget: upper part
2. constraint gadget: lower part

**Tripartite Matching:** Given  $T \subseteq B \times G \times H$ ,  
 $|B| = |G| = |H| = n$ , try to find  $n$  triples in  $T$  s.t. no two of  
which have a component in common.

(B: boys, G: girls, H: homes)

**Theorem 9.8** TRIPARTITE MATCHING is NP-complete.

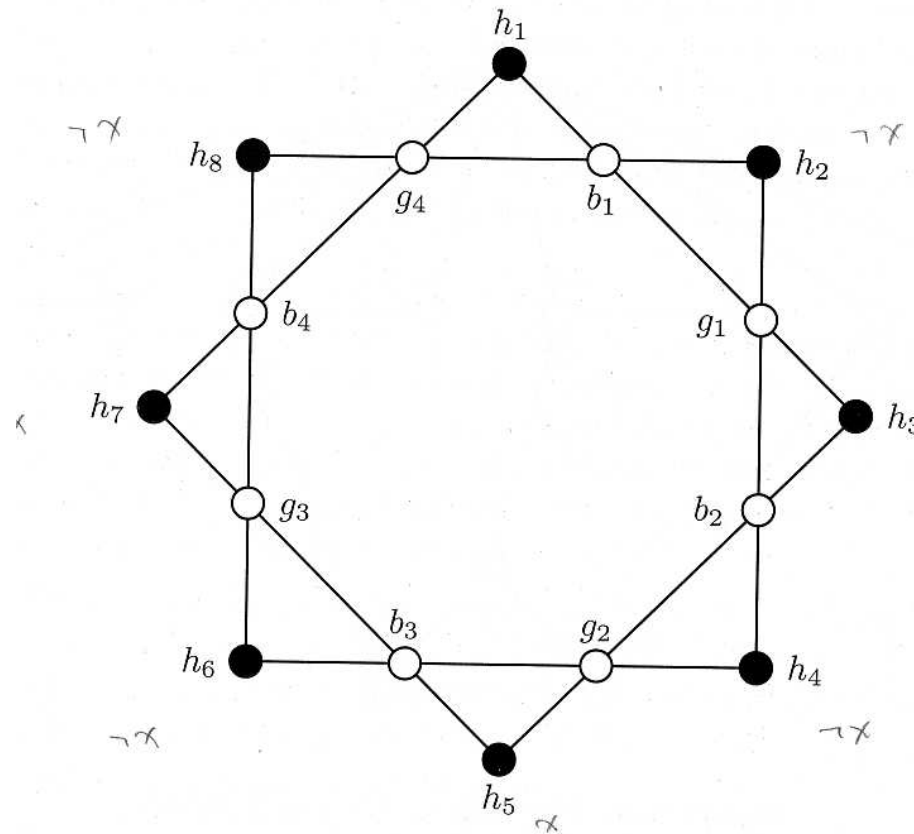


Figure 9-9. The choice-consistency gadget.

Reduce 3-SAT to it.

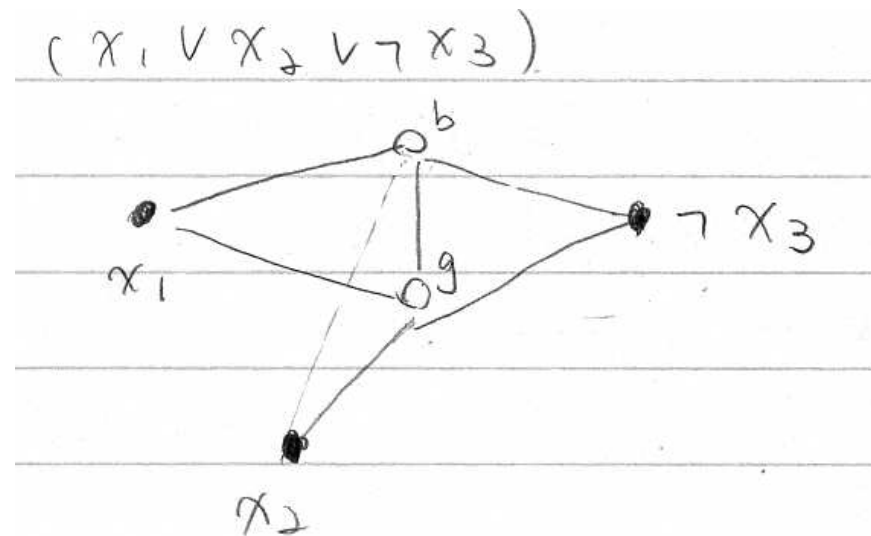
1. For each variable  $x_i$ , we construct a choice-consistency

gadget.

(a) Let  $k$  be the maximum of the occurrence of  $x$  and the occurrence of  $\neg x$ .

(b) There are  $k$  boys,  $k$  girls,  $2k$  homes in this gadget.

2. For each clause  $(\alpha \vee \beta \vee \gamma)$ , construct a triple  $(b, g, h)$  where  $h$  is either  $\alpha, \beta$ , or  $\gamma$ , not joined by another triple in this step.



3. Suppose there are  $m$  clauses. Then there are at least  $3m$  homes. The number of boys is  $\frac{|H|}{2} + m \leq |H|$ . Introduce  $l$  more boys & girls such that  $|B| = |G| = |H|$ . For each of the  $l$  boys and girls, add  $|H|$  triples that connect to all homes.

**Set Covering:**  $F = \{S_1, \dots, S_m\}$  of subsets of a finite set  $U$ . Find a minimum sets in  $F$  whose union is  $U$ .

**Set Packing:**  $F = \{S_1, \dots, S_m\}$  of subsets of a finite set  $U$ . Find a maximum sets in  $F$  that are pairwise disjoint.

**Exact Cover By 3-Set:**  $F = \{S_1, \dots, S_m\}$  of subsets of a finite set  $U$ , and  $|S_i| = 3$ ,  $|U| = 3m$  for some  $m$ . Find  $m$  sets in  $F$  that are disjoint and have  $U$  as their union.

All of these problems are generalization of TRIPARTITE MATCHING. Hence, they are all NP-complete.

SET COVERING

SET PACKING

EXACT COVER BY 3-SET

TRIPARTITE MATCHING

↑ generalize

↓ specialize

**Integer Programming:** Given a system of linear inequalities with integer coefficients, does it have an integer solution?

**Theorem** INTEGER PROGRAMMING is NP-complete.

Reduce SET COVERING to it. Let  $F = \{S_1, \dots, S_n\}$  be subsets of  $U$ .  $x = (x_1 \ x_2 \ \cdots \ x_n)^t$ .

$$x_i = \begin{cases} 1 & \text{if } S_i \text{ is in the cover;} \\ 0 & \text{otherwise.} \end{cases}$$



$A = (a_{i,j})$ ,  $a_{i,j} = 1$  iff the  $i$ th element in  $U$  belongs to  $S_j$ .

$$\Rightarrow \begin{cases} Ax \geq 1; \\ \sum_{i=1}^n x_i \leq B, \text{ where } B \text{ is the budget;} \\ 0 \leq x_i \leq 1. \end{cases}$$

**Knapsack:**  $\{1, 2, \dots, n\}$ ,  $n$  items. Item  $i$  has value  $v_i > 0$  and weight  $w_i > 0$ . Try to find a subset  $S \subseteq \{1, \dots, n\}$  such that  $\sum_{i \in S} w_i \leq W$  and  $\sum_{i \in S} v_i \geq K$  for some  $W$  and  $K$ .

**Theorem 9.10** KNAPSACK is NP-complete.

→	0	0	0	1	0	1	1	0	0	0	0	0
	1	1	0	0	1	0	0	0	0	0	0	0
→	1	0	1	0	0	0	0	1	0	0	0	0
→	0	1	0	0	0	0	0	0	1	0	0	1
	0	0	1	1	1	0	0	0	0	0	0	0
→	0	0	0	0	1	0	0	0	0	1	1	0
+	1	0	1	1	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1

**Figure 9.10.** Reduction to KNAPSACK.

Reduce EXACT COVER BY 3-SET to it.  $\{S_1, S_2, \dots, S_n\}$ , an instance of EXACT COVER BY 3-SET,  $U = \{1, 2, \dots, 3m\}$ .

Let  $v_i = w_i = \sum_{j \in S_i} (n+1)^{3m-i}$  and

$$W = K = \sum_{j=0}^{3m-1} (n+1)^j.$$

**Proposition 9.4** Any instance of KNAPSACK can be solved in  $O(nW)$  time, where  $n$  is the number of items and  $W$  is the weight limit.

We can solve this by [dynamic programming](#).

$V(w, i)$ : the largest value attainable by selecting some among the  $i$  first items so that the total weight is exactly  $w$ .

$$\begin{cases} V(w, i + 1) = \max\{V(w, i), v_{i+1} + V(w - w_{i+1}, i)\}; \\ V(w, 0) = 0. \end{cases}$$

If  $V(W, n) \geq K$ , then answer “yes.”