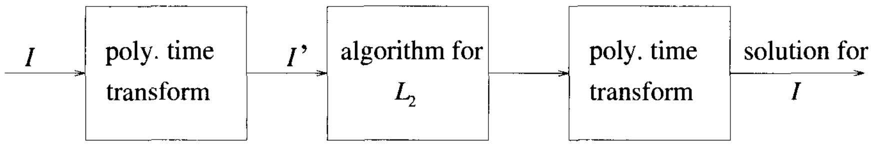

Figure 11.3 Reduction of L_1 to L_2

(e) $Y :=$ Choice$(0, 1)$;

(f) $Y :=$ Choice(r, u);, where r and u are variables

2. Show how to encode the following instructions as CNF formulas: (a) **for** and (b) **while**.

3. Prove or disprove: If there exists a polynomial time algorithm to convert a boolean formula in CNF into an equivalent formula in DNF, then $\mathcal{P} = \mathcal{NP}$.

11.3 $\mathcal{NP}$-HARD GRAPH PROBLEMS

The strategy we adopt to show that a problem L_2 is $\mathcal{NP}$-hard is:

1. Pick a problem L_1 already known to be $\mathcal{NP}$-hard.

2. Show how to obtain (in polynomial deterministic time) an instance I' of L_2 from any instance I of L_1 such that from the solution of I' we can determine (in polynomial deterministic time) the solution to instance I of L_1 (see Figure 11.3).

3. Conclude from step (2) that $L_1 \propto L_2$.

4. Conclude from steps (1) and (3) and the transitivity of $\propto$ that L_2 is $\mathcal{NP}$-hard.

For the first few proofs we go through all the above steps. Later proofs explicitly deal only with steps (1) and (2). An $\mathcal{NP}$-hard decision problem L_2 can be shown to be $\mathcal{NP}$-complete by exhibiting a polynomial time nondeterministic algorithm for L_2. All the $\mathcal{NP}$-hard decision problems we deal with here are $\mathcal{NP}$-complete. The construction of polynomial time nondeterministic algorithms for these problems is left as an exercise.

11.3.1 Clique Decision Problem (CDP)

The clique decision problem was introduced in Section 11.1. We show in Theorem 11.2 that CNF-satisfiability $\propto$ CDP. Using this result, the transitivity of $\propto$, and the knowledge that satisfiability $\propto$ CNF-satisfiability (Section 11.2), we can readily establish that satisfiability $\propto$ CDP. Hence, CDP is $\mathcal{NP}$-hard. Since, CDP $\in \mathcal{NP}$, CDP is also $\mathcal{NP}$-complete.

Theorem 11.2 CNF-satisfiability $\propto$ clique decision problem.

Proof: Let $F = \bigwedge_{1 \le i \le k} C_i$ be a propositional formula in CNF. Let x_i, $1 \le i \le n$, be the variables in F. We show how to construct from F a graph $G = (V, E)$ such that G has a clique of size at least k if and only if F is satisfiable. If the length of F is m, then G is obtainable from F in $O(m)$ time. Hence, if we have a polynomial time algorithm for CDP, then we can obtain a polynomial time algorithm for CNF-satisfiability using this construction.

For any F, $G = (V, E)$ is defined as follows: $V = \{\langle\sigma, i\rangle | \sigma$ is a literal in clause $C_i\}$ and $E = \{(\langle\sigma, i\rangle, \langle\delta, j\rangle) \mid i \ne j$ and $\sigma \ne \bar{\delta}\}$. A sample construction is given in Example 11.11.

Claim: F is satisfiable if and only if G has a clique of size $\ge k$.

Proof of Claim: If F is satisfiable, then there is a set of truth values for x_i, $1 \le i \le n$, such that each clause is true with this assignment. Thus, with this assignment there is at least one literal σ in each C_i such that σ is true. Let $S = \{\langle\sigma, i\rangle \mid \sigma$ is true in $C_i\}$ be a set containing exactly one $\langle\sigma, i\rangle$ for each i. Between any two nodes $\langle\sigma, i\rangle$ and $\langle\delta, j\rangle$ in S there is an edge in G, since $i \ne j$ and both σ and δ have the value true. Thus, S forms a clique in G of size k.

Similarly, if G has a clique $K = (V', E')$ of size at least k, then let $S = \{\langle\sigma, i\rangle \mid \langle\sigma, i\rangle \in V'\}$. Clearly, $|S| = k$ as G has no clique of size more than k. Furthermore, if $S' = \{\sigma \mid \langle\sigma, i\rangle \in S$ for some $i\}$, then S' cannot contain both a literal δ and its complement $\bar{\delta}$ as there is no edge connecting $\langle\delta, i\rangle$ and $\langle\bar{\delta}, j\rangle$ in G. Hence by setting $x_i =$ true if $x_i \in S'$ and $x_i =$ false if $\bar{x}_i \in S'$ and choosing arbitrary truth values for variables not in S', we can satisfy all clauses in F. Hence, F is satisfiable iff G has a clique of size at least k. □

Example 11.11 Consider $F = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$. The construction of Theorem 11.2 yields the graph of Figure 11.4. This graph contains six cliques of size two. Consider the clique with vertices $\{\langle x_1, 1\rangle, \langle\bar{x}_2, 2\rangle\}$. By setting $x_1 =$ true and $\bar{x}_2 =$ true (that is, $x_2 =$ false), F is satisfied. The x_3 may be set either to true or false. □

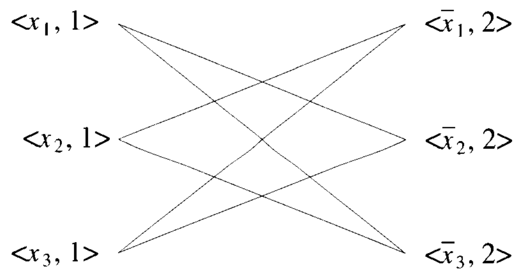

Figure 11.4 A sample graph and satisfiability

11.3.2 Node Cover Decision Problem (NCDP)

A set $S \subseteq V$ is a *node cover* for a graph $G = (V, E)$ if and only if all edges in E are incident to at least one vertex in S. The size $|S|$ of the cover is the number of vertices in S.

Example 11.12 Consider the graph of Figure 11.5. $S = \{2, 4\}$ is a node cover of size 2. $S = \{1, 3, 5\}$ is a node cover of size 3. □

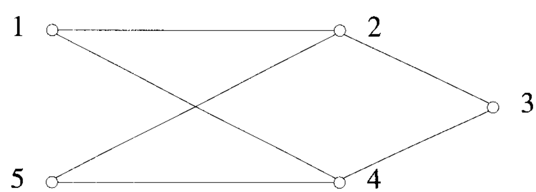

Figure 11.5 A sample graph and node cover

In the node cover decision problem we are given a graph G and an integer k. We are required to determine whether G has a node cover of size at most k.

Theorem 11.3 The clique decision problem $\propto$ the node cover decision problem.

Proof: Let $G = (V, E)$ and k define an instance of CDP. Assume that $|V| = n$. We construct a graph G' such that G' has a node cover of size at most $n - k$ if and only if G has a clique of size at least k. Graph G' is given by $G' = (V, \bar{E})$, where $\bar{E} = \{(u, v) \mid u \in V, v \in V \text{ and } (u, v) \notin E\}$. The set G' is known as the *complement* of G.

Now, we show that G has a clique of size at least k if and only if G' has a node cover of size at most $n - k$. Let K be any clique in G. Since there are no edges in $\bar{E}$ connecting vertices in K, the remaining $n - |K|$ vertices in G' must cover all edges in $\bar{E}$. Similarly, if S is a node cover of G', then $V - S$ must form a complete subgraph in G.

Since G' can be obtained from G in polynomial time, CDP can be solved in polynomial deterministic time if we have a polynomial time deterministic algorithm for NCDP. □

Example 11.13 Figure 11.6 shows a graph G and its complement G'. In this figure, G' has a node cover of $\{4, 5\}$, since every edge of G' is incident either on the node 4 or on the node 5. Thus, G has a clique of size $5 - 2 = 3$ consisting of the nodes $1, 2,$ and 3. □

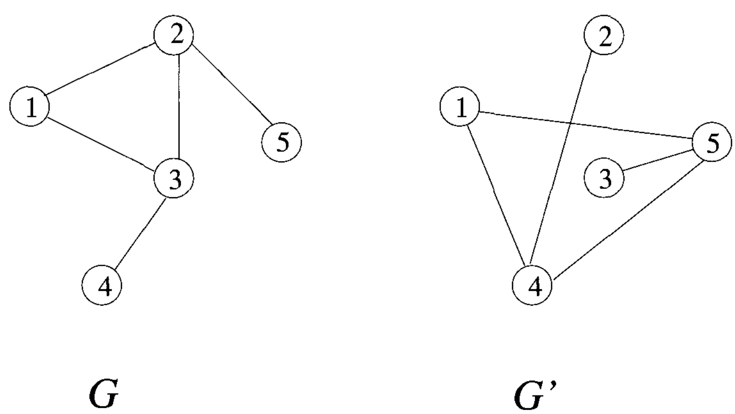

Figure 11.6 A graph and its complement

Note that since CNF-satisfiability $\propto$ CDP, CDP $\propto$ NCDP and $\propto$ is transitive, it follows that NCDP is $\mathcal{NP}$-hard. NCDP is also in $\mathcal{NP}$ because we can nondeterministically choose a subset $C \subseteq V$ of size k and verify in polynomial time that C is a cover of G. So NCDP is $\mathcal{NP}$-complete.

11.3.3 Chromatic Number Decision Problem (CNDP)

A coloring of a graph $G = (V, E)$ is a function $f : V \rightarrow \{1, 2, \ldots, k\}$ defined for all $i \in V$. If $(u, v) \in E$, then $f(u) \neq f(v)$. The *chromatic number decision problem* is to determine whether G has a coloring for a given k.

Example 11.14 A possible 2-coloring of the graph of Figure 11.5 is $f(1) = f(3) = f(5) = 1$ and $f(2) = f(4) = 2$. Clearly, this graph has no 1-coloring. □

In proving CNDP to be $\mathcal{NP}$-hard, we shall make use of the $\mathcal{NP}$-hard problem SATY. This is the CNF-satisfiability problem with the restriction that each clause has at most three literals. The reduction CNF-satisfiability $\propto$ SATY is left as an exercise.

Theorem 11.4 Satisfiability with at most three literals per clause $\propto$ chromatic number decision problem.

Proof: Let F be a CNF formula having at most three literals per clause and having r clauses $C_1, C_2, \ldots, C_r$. Let x_i, $1 \leq i \leq n$, be the n variables in F. We can assume $n \geq 4$. If $n < 4$, then we can determine whether F is satisfiable by trying out all eight possible truth value assignments to x_1, x_2, and x_3. We construct, in polynomial time, a graph G that is $n + 1$ colorable if and only if F is satisfiable. The graph $G = (V, E)$ is defined by

$$V = \{x_1, x_2, \ldots, x_n\} \cup \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n\} \cup \{y_1, y_2, \ldots, y_n\} \cup \{C_1, C_2, \ldots, C_r\}$$

where $y_1, y_2, \ldots, y_n$ are new variables, and

$$E = \{(x_i, \quad \bar{x}_i), 1 \leq i \leq n\} \cup \{(y_i, y_j) | i \neq j\} \cup \{(y_i, x_j) | i \neq j\}$$
$$\cup \{(y_i, \bar{x}_j) | i \neq j\} \cup \{(x_i, C_j) | x_i \notin C_j\} \cup \{\bar{x}_i, C_j) | \bar{x}_i \notin C_j\}$$

To see that G is $n + 1$ colorable if and only if F is satisfiable, we first observe that the y_i's form a complete subgraph on n vertices. Hence, each y_i must be assigned a distinct color. Without loss of generality we can assume that in any coloring of G, y_i is given the color i. Since y_i is also connected to all the x_j's and $\bar{x}_j$'s except x_i and $\bar{x}_i$, the color i can be assigned to only x_i and $\bar{x}_i$. However, $(x_i, \bar{x}_i) \in E$ and so a new color, $n + 1$, is needed for one of these vertices. The vertex that is assigned the new color $n + 1$ is called a *false vertex*. The other vertex is a *true* vertex. The only way to color G using $n + 1$ colors is to assign color $n + 1$ to one of $\{x_i, \bar{x}_i\}$ for each i, $1 \leq i \leq n$.

Under what conditions can the remaining vertices be colored using no new colors? Since $n \geq 4$ and each clause has at most three literals, each C_i is adjacent to a pair of vertices $x_j, \bar{x}_j$ for at least one j. Consequently, no C_i can be assigned the color $n+1$. Also, no C_i can be assigned a color corresponding to an x_j or $\bar{x}_j$ not in clause C_i. The last two statements imply that the only colors that can be assigned to C_i correspond to vertices x_j or $\bar{x}_j$ that are in clause C_i and are true vertices. Hence, G is $n+1$ colorable if and only if there is a true vertex corresponding to each C_i. So, G is $n+1$ colorable iff F is satisfiable. □

11.3.4 Directed Hamiltonian Cycle (DHC) (*)

A directed Hamiltonian cycle in a directed graph $G = (V, E)$ is a directed cycle of length $n = |V|$. So, the cycle goes through every vertex exactly once and then returns to the starting vertex. The DHC problem is to determine whether G has a directed Hamiltonian cycle.

Example 11.15 1, 2, 3, 4, 5, 1 is a directed Hamiltonian cycle in the graph of Figure 11.7. If the edge $\langle 5, 1\rangle$ is deleted from this graph, then it has no directed Hamiltonian cycle. □

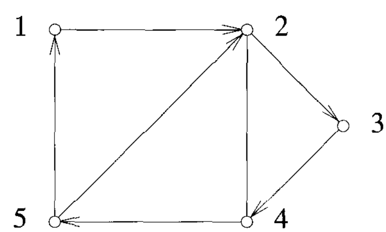

Figure 11.7 A sample graph and Hamiltonian cycle

Theorem 11.5 CNF-satisfiability $\propto$ directed Hamiltonian cycle.

Proof: Let F be a propositional formula in CNF. We show how to construct a directed graph G such that F is satisfiable if and only if G has a directed Hamiltonian cycle. Since this construction can be carried out in time polynomial in the size of F, it will follow that CNF-satisfiability $\propto$ DHC. Understanding the construction of G is greatly facilitated by the use of an example. The example we use is $F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$, where

$$\begin{aligned}
C_1 &= x_1 \vee \bar{x}_2 \vee x_4 \vee \bar{x}_5 \\
C_2 &= \bar{x}_1 \vee x_2 \vee x_3 \\
C_3 &= \bar{x}_1 \vee \bar{x}_3 \vee x_5 \\
C_4 &= \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4 \vee \bar{x}_5
\end{aligned}$$

Assume that F has r clauses $C_1, C_2, \ldots, C_r$ and n variables $x_1, x_2, \ldots, x_n$. Draw an array with r rows and $2n$ columns. Row i denotes clause C_i. Each variable x_i is represented by two adjacent columns, one for each of the literals x_i and $\bar{x}_i$. Figure 11.8 shows the array for the example formula. Insert a $\odot$ into column x_i and row C_j if and only if x_i is a literal in C_j. Insert a $\odot$ into column $\bar{x}_i$ and row C_j if and only if $\bar{x}_i$ is a literal in C_j. Between each pair of columns x_i and $\bar{x}_i$ introduce two vertices u_i and v_i, u_i at the top and v_i at the bottom of the column. For each i, draw two chains of edges upward from v_i to u_i, one connecting together all $\odot$s in column x_i and the other connecting all $\odot$s in column $\bar{x}_i$ (see Figure 11.8). Now, draw edges $\langle u_i, v_{i+1} \rangle$, $1 \le i < n$. Introduce a box $\boxed{\text{i}}$ at the right end of each row C_i, $1 \le i < r$. Draw the edges $\langle u_n, \boxed{1} \rangle$ and $\langle \boxed{\text{r}}, v_1 \rangle$. Draw edges $\langle \boxed{\text{i}}, \boxed{\text{i+1}} \rangle$, $1 \le i < r$ (see Figure 11.8).

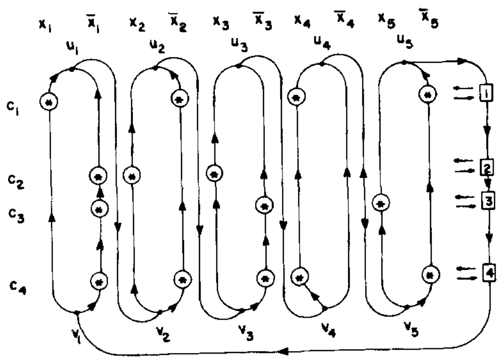

Figure 11.8 Array structure for the formula in Theorem 11.5

To complete the graph, we replace each $\odot$ and $\boxed{\text{i}}$ by a subgraph. Each $\odot$ is replaced by the subgraph of Figure 11.9(a) (of course, unique vertex labelings are needed for each copy of the subgraph). Each box $\boxed{\text{i}}$ is replaced by the subgraph of Figure 11.10. In this subgraph A_i is an entrance node and B_i an exit node. The edges $\langle \boxed{\text{i}}, \boxed{\text{i+1}} \rangle$ referred to earlier are really $\langle B_i, A_{i+1} \rangle$. Edge $\langle u_n, \boxed{1} \rangle$ is $\langle u_n, A_1 \rangle$ and $\langle \boxed{\text{r}}, v_1 \rangle$ is $\langle B_r, v_1 \rangle$. The variable j_i is the number of literals in clause C_i. In the subgraph of Figure 11.10 an edge of the type shown in Figure 11.11 indicates a connection to a $\odot$ subgraph in row C_i. $R_{i,a}$ is connected to the 1 vertex of the $\odot$ and $R_{i,a+1}$ (or $R_{i,1}$ if $a = j_i$) is entered from the 3 vertex.

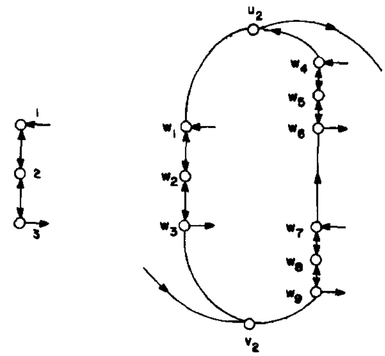

Figure 11.9 The $\odot$ subgraph and its insertion into column 2

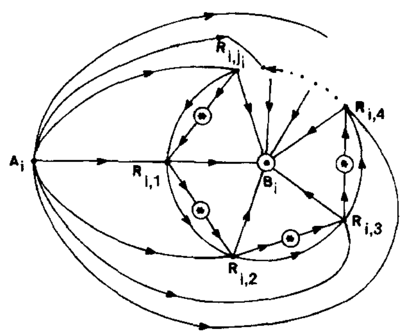

Figure 11.10 The H_i subgraph

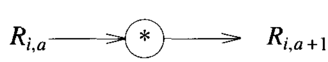

Figure 11.11 A construct in the proof of Theorem 11.5

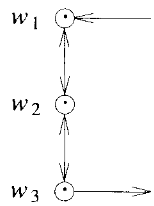

Figure 11.12 Another construct in the proof of Theorem 11.5

Thus in the $\odot$ subgraph (shown in Figure 11.12) of Figure 11.9(b) w_1 and w_3 are the 1 and 3 vertices respectively. The incoming edge is $\langle R_{i,1}, w_1\rangle$ and the outgoing edge is $\langle w_3, R_{i,2}\rangle$. This completes the construction of G.

If F is satisfiable, then let S be an assignment of truth values for which F is true. A Hamiltonian cycle for G can start at v_1 and go to u_1, then to v_2, then to u_2, then to v_3, then to $u_3, \ldots$, and then to u_n. In going from v_i to u_i, this cycle uses the column corresponding to x_i if x_i is true in S. Otherwise it goes up the column corresponding to $\bar{x}_i$. From u_n this cycle goes to A_1 and then through $R_{1,1}, R_{1,2}, R_{1,3}, \ldots, R_{1,j_i}$, and B_1 to A_2 to $\cdots$ to v_1. In going from $R_{i,a}$ to $R_{i,a+1}$ in any subgraph $\boxed{\text{i}}$, a diversion is made to a $\odot$ subgraph in row i if and only if the vertices of that $\odot$ subgraph are not already on the path from v_i to $R_{i,a}$. Note that if C_i has i_j literals, then the construction of $\boxed{\text{i}}$ allows a diversion to at most $i_j - 1$ $\odot$ subgraphs. This is adequate as at least one $\odot$ subgraph must already have been traversed in row C_i (because at least one such subgraph must correspond to a true literal). So, if F is satisfiable, then G has a directed Hamiltonian cycle.

It remains to show that if G has a directed Hamiltonian cycle, then F is satisfiable. This can be seen by starting at vertex v_1 on any Hamiltonian cycle for G. Because of the construction of the $\odot$ and $\boxed{\text{i}}$ subgraphs, such a cycle must proceed by going up exactly one column of each pair $(x_i, \bar{x}_i)$. In addition, this part of the cycle must traverse at least one $\odot$ subgraph in each row. Hence the columns used in going from v_i to u_i, $1 \le i \le n$, define a truth assignment for which F is true.

We conclude that F is satisfiable if and only if G has a Hamiltonian cycle. The theorem now follows from the observation that G can be obtained from F in polynomial time. □

11.3.5 Traveling Salesperson Decision Problem (TSP)

The traveling salesperson problem was introduced in Chapter 5. The corresponding decision problem is to determine whether a complete directed

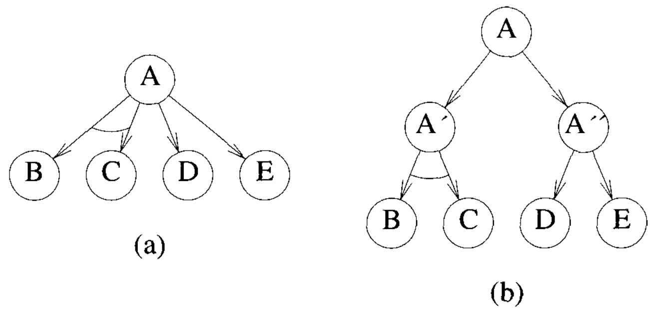

Figure 11.13 Graphs representing problems

graph $G = (V, E)$ with edge costs $c(u, v)$ has a tour of cost at most M.

Theorem 11.6 Directed Hamiltonian cycle (DHC) $\propto$ the traveling salesperson decision problem (TSP).

Proof: From the directed graph $G = (V, E)$ construct the complete directed graph $G' = (V, E')$, $E' = \{\langle i, j\rangle \mid i \neq j\}$ and $c(i, j) = 1$ if $\langle i, j\rangle \in E$; $c(i, j) = 2$ if $i \neq j$ and $\langle i, j\rangle \notin E$. Clearly, G' has a tour of cost at most n iff G has a directed Hamiltonian cycle. □

11.3.6 AND/OR Graph Decision Problem (AOG)

Many complex problems can be broken down into a series of subproblems such that the solution of all or some of these results in the solution of the original problem. These subproblems can be broken down further into sub-subproblems, and so on, until the only problems remaining are sufficiently primitive as to be trivially solvable. This breaking down of a complex problem into several subproblems can be represented by a directed graphlike structure in which nodes represent problems and descendents of nodes represent the subproblems associated with them.

Example 11.16 The graph of Figure 11.13(a) represents a problem A that can be solved by solving either both the subproblems B and C or the single subproblem D or E. □

Groups of subproblems that must be solved in order to imply a solution to the parent node are joined together by an arc going across the respective edges (as the arc across the edges $\langle A, B\rangle$ and $\langle A, C\rangle$). By introducing dummy