Recall the rook game from Example 10.3.5. In Exercise 4 we explicitly compute the set of winning positions of Player I, and Exercise 5 considers a variant of the game. We keep the restriction that in each move the piece has to be moved, that it cannot be moved east or north, and that the player who moves it to the lower-left square wins.

4S. Show formally that the set ROOK $= \{(n,m) \mid$ Player I can force a win with the rook starting in $(n,m)\}$ equals $\{(n,m) \mid n \neq m\}$. *Hint:* Find a winning strategy for player I.

5. Assume instead of a rook we use the king (who can move one square in any direction). Determine the set KING $= \{(n,m) \mid$ Player I can force a win with the king starting in $(n,m)\}$. What is the winning strategy for player I?

6. Show that, if we extend our notion of a linear program (see Example 10.3.6) to also include equalities $(a_{i,1}x_1 + a_{i,2}x_2 + \ldots + a_{i,n}x_n = b_n)$, the problem can still be solved in polynomial time.

7S. Complete the proof of Theorem 10.3.15 by showing what to do with a clause $\ell_1 \vee \ell_2$ containing two literals.

8. Let CYCLE $= \{(G,k) \mid G$ contains a simple cycle of length at least $k\}$. Show that the Hamiltonian-cycle problem reduces to CYCLE.

10.4 NP-Complete Problems

Using the **NP**-completeness of SAT and 3SAT as a starting point, we show that finding an independent set or a clique or coloring a graph are hard tasks to solve. These are fundamental problems, which have been used in the **NP**-completeness proofs of many other problems.

Independent Sets and Cliques

A set of vertices U is **independent** in a graph $G = (V,E)$ if there is no edge incident on two vertices of U. The size of a maximal independent set in G is called the **independence number** of G and written $\alpha(G)$.

Example 10.4.1. Consider the graph in Figure 10.4.1. It contains an independent set of size 3 consisting of nodes 1, 3, and 6. The set is independent, since no two of the nodes are adjacent in the graph. To show that the independence number of the graph is 3, we have to show that there is no independent set of 4 vertices. If there were a set of four independent vertices, it could contain at most one of the vertices 1 and 2; hence three vertices from the set $\{3,4,5,6\}$ need to be independent, which is not possible.

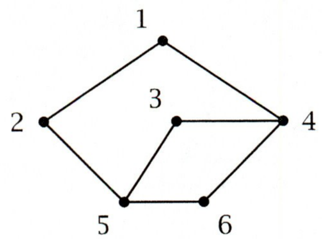

Figure 10.4.1 A graph of independence number 3. □

The independent-set decision problem can be phrased as follows:

Independent Set
Instance: Graph $G = (V, E)$, integer $k \leq |V|$.
Question: Is $\alpha(G) \geq k$; that is, does G contain an independent set of size at least k?

Theorem 10.4.2. *The independent-set problem is* **NP**-*complete.*

Proof. We leave it to Exercise 10.31 to show that the independent-set problem is in **NP**. By Theorem 10.3.10, we can show that the independent-set problem is **NP**-complete by reducing the **NP**-complete problem 3SAT to it.

Suppose that we are given a Boolean formula φ in 3-CNF with clauses $C_1, \ldots, C_m$. We build a graph $G = (V, E)$ such that φ is satisfiable if and only if G contains an independent set of size m, the number of clauses. For each clause C_i we draw a triangle (three vertices connected by three edges) and label the vertices of each triangle with the literals in the corresponding clause. We call two literals **contradictory** if they are the positive and negative versions of the same variable. For example, $\overline{x}_3$ and x_3 are contradictory, but $\overline{x}_3$ and x_1 are not. To the triangles, we add edges between pairs of vertices if they are labeled by contradictory literals. We claim that the resulting graph G contains an independent set of size m if and only if φ is satisfiable.

For example, Figure 10.4.2 shows the graph G resulting from

$$\varphi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_2 \vee x_3 \vee \overline{x}_4) \wedge (\overline{x}_1 \vee x_2 \vee x_4) \wedge (x_2 \vee \overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee x_4).$$

The graph G contains an independent set on five vertices as shown by the circles in Figure 10.4.2: $\overline{x}_2$ from C_1, x_3 from C_2, $\overline{x}_1$ from C_3, $\overline{x}_4$ from C_4, and $\overline{x}_1$ from C_5. This corresponds to the truth assignment that makes x_1, x_2, and x_4 false and x_3 true. That truth assignment satisfies φ, since each clause is satisfied.

To prove the claim that G contains an independent set of size m if and only if φ is satisfiable, we first assume that G contains an independent set of size m. Fix such an independent set U of m vertices. Then, U can contain at most one vertex from each triangle, since there are edges between any two vertices of the same triangle. Since there are only m triangles altogether, U must contain exactly one vertex from each triangle. Consider the labels

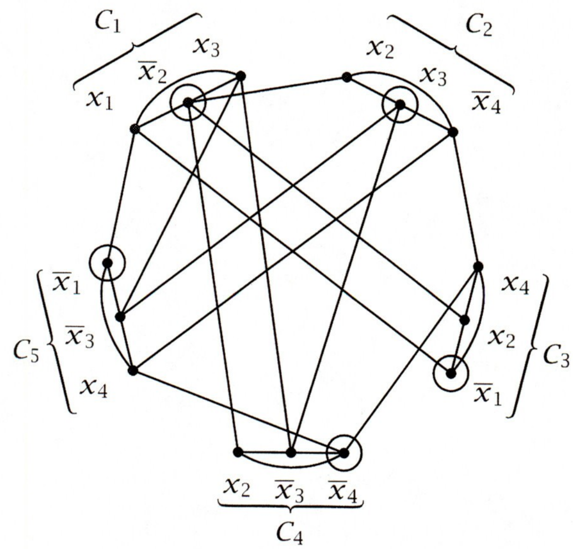

Figure 10.4.2 The graph G for $\varphi = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_2 \vee x_3 \vee \overline{x}_4) \wedge (\overline{x}_1 \vee x_2 \vee x_4) \wedge (x_2 \vee \overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee x_4)$. Each clause corresponds to a (flattened) triangle. The five nodes within circles form a maximal independent set.

associated with the vertices in U. Since U is an independent set, no two of these labels are contradictory. Therefore, if we assign the value "true" to the literals associated with a vertex in U, we obtain a partial truth assignment to the variables in φ. This truth assignment satisfies φ, since it makes at least one literal in each clause true. The truth assignment still satisfies φ after we assign arbitrary truth values to any remaining variables.

It remains to show that if φ is satisfiable, then G contains an independent set of size at least m. Assume then that φ is satisfiable and fix a truth assignment that satisfies φ. With this truth assignment, at least one literal in each clause of φ has to be made true by what it means to be satisfiable. Choose one such literal from each clause and collect the corresponding vertices in a set U. Then U contains m vertices, since we picked one vertex for each clause (from each triangle). We claim that there cannot be any edge between two vertices of U. Such an edge would connect to two contradictory literals, which is not possible because we chose the vertices according to a truth assignment and it is not possible for both a variable and its negation to be true. Therefore, U is an independent set of size m. ■

A property related to being an independent set is that of being a clique. A set of vertices U is a **clique** in $G = (V, E)$ if all edges between any two vertices in U belong to E. The size of a maximal clique is called the **clique number** of the graph and written $\omega(G)$.

Clique
Instance: Graph $G = (V, E)$, integer $k \leq |V|$.
Question: Is $\omega(G) \geq k$; that is, does G contain a clique of size at least k?

Cliques are areas of high connectivity in a graph. Depending on the context, high connectivity can be desirable or undesirable. When we think of the graph as a network, cliques are highly connected subgroups of the network, very robust against server or network failures and, therefore, desirable. If, on the other hand, the graph models conflicts and we have to find a coloring of the graph, then large cliques are undesirable since they force us to use many colors: Each vertex of a clique requires a different color (resource). In either case, it is often important to know the size of a maximal clique in a graph.

Solving the clique problem is as hard as solving the independent-set problem, since an independent set in G is a clique in $\overline{G}$, the complement of G (for the definition of $\overline{G}$, see Section 2.5).

Theorem 10.4.3. *The clique problem is* **NP**-*complete.*

Proof. We leave it to Exercise 10.32 to show that the clique problem is in **NP**. Because of Theorem 10.3.10, we can prove that the clique problem is **NP**-complete by reducing any **NP**-complete problem to it. We show how to reduce the **NP**-complete independent-set problem to the clique problem. The reduction maps an instance G, k of the independent-set problem to the instance $\overline{G}, k$. To show that this function is indeed a reduction, assume that G contains an independent set U on at least k vertices. Since U is an independent set in G, G does not contain any edges between any pair of vertices in U. Then, by definition, $\overline{G}$ contains all edges between any two vertices of U; hence, U is a clique on k vertices in $\overline{G}$ (Figure 10.4.3).

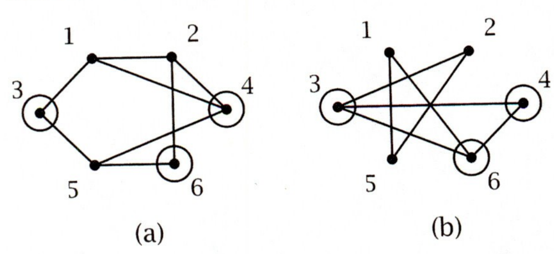

Figure 10.4.3 A graph G in (a) and its complement $\overline{G}$ in (b) with independent set $\{3, 4, 6\}$ in G and the corresponding clique $\{3, 4, 6\}$ in $\overline{G}$.

It remains to show that if $\overline{G}$ contains a clique on k vertices, then G has an independent set of size k. Assume that $\overline{G}$ contains a clique U on k vertices. Then $\overline{G}$ contains all edges between any two vertices of U. Again, by definition of the complement of a graph, G does not contain any edges between any two vertices of U and therefore U is an independent set of size at least k in G. In summary, we have shown that G contains an independent set of size at least k if and only if $\overline{G}$ contains a clique of size at least k, showing that the function that maps G, k to $\overline{G}, k$ is a reduction from the independent-set problem to the clique problem. Therefore, the clique problem is **NP**-complete. ■

Graph Coloring

In Theorem 10.2.16 we showed that graph k-colorability is in **NP**. In case $k = 2$, we can actually solve the problem in polynomial time using breadth-first search (see Exercise 10.1). The problem turns out to be **NP**-complete for any $k \geq 3$.

Theorem 10.4.4. *Graph 3-colorability is* **NP**-*complete.*

Proof. Since we already know that graph 3-colorability is in **NP**, it is sufficient to show how to reduce 3SAT to 3-colorability. We translate the elements of the 3SAT problem, variables and clauses, into components in the graph that simulate the satisfiability behavior in the context of graph coloring. Such components are usually called *gadgets*, and this type of construction is called a *gadget construction*.

We are given a formula φ in 3-CNF with clauses $C_1, \ldots, C_m$. Our goal is to construct a graph G that is 3-colorable if and only if φ is satisfiable. We build G step by step. We start with a triangle on three vertices that we name b, t, and f as shown in Figure 10.4.4. Since there are only three colors available, b, t, and f have to be assigned those three different colors.

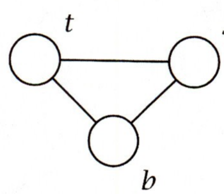

Figure 10.4.4 The base triangle $\{b, t, f\}$.

The colors of t and f correspond to true and false, while the third color, the color of b, is used for encoding purposes. For each variable x in φ, we take two new vertices v_x and $v_{\overline{x}}$ to build another triangle with b at the base (see Figure 10.4.5).

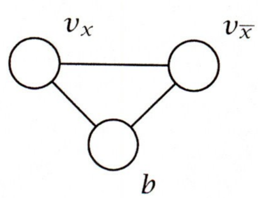

Figure 10.4.5 The triangle $\{v_x, v_{\overline{x}}, b\}$. The vertex b is the same vertex we saw in Figure 10.4.4.

This triangle forces v_x and $v_{\overline{x}}$ to be colored with the colors of t and f, and since the colors of v_x and $v_{\overline{x}}$ have to be different (because of the edge between them), we can read a coloring as a truth assignment to x; namely, if v_x has the same color as t, we can call x true and false otherwise. In this way, a 3-coloring of the graph induces a truth assignment to the variables, and we therefore speak about a coloring satisfying φ.

For each clause $C_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$, we build a gadget that guarantees that at least one literal in each clause is satisfied by the coloring. The gadget is displayed in Figure 10.4.6. We call the six vertices in a row the *baseline* of the gadget and the outer two vertices of the baseline its *endpoints*. The three labeled vertices are the tops of the triangles. This completes the construction of G.

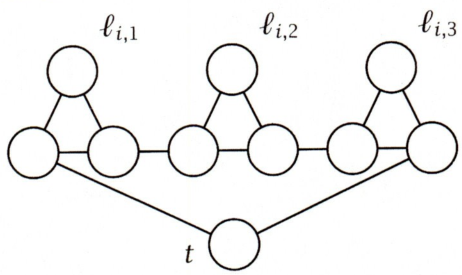

Figure 10.4.6 The clause gadget for 3-colorings. The vertex t is the same vertex we saw in Figure 10.4.4, and the vertices labeled $\ell_{i,j}$ are the vertices we created for each variable.

Figure 10.4.7 shows G for the formula $\varphi = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee x_3)$.

We first argue that a 3-coloring of G translates to a satisfying assignment of φ. The vertices t, f, and b have to be colored with the three different colors; so without loss of generality let us assume that t is red, f is green, and b is blue. Now v_x and $v_{\overline{x}}$ have to be colored red and green, or green and red, since there is an edge between them, and both are connected to the blue b. From this information we construct a truth assignment as follows. If v_x is red, we let x be true, and false otherwise. Fix this truth assignment. We show that this assignment satisfies φ. Consider a clause $C_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$ and its corresponding gadget. Assume that $\ell_{i,1}$, $\ell_{i,2}$, and $\ell_{i,3}$ are all false under the truth assignment we fixed. Then the tops of all three triangles in the gadgets are green. This forces the vertices in the baseline of the gadget to be alternately blue and red; since there are six vertices in the baseline, one of the endpoints has to be red, which is not possible since it is connected to t, which is colored red (see Figure 10.4.8).

Hence, one of the tops is colored red, meaning that the corresponding literal is true in the truth assignment we fixed, and therefore clause C_i is satisfied. Since this argument is true for all clauses, all clauses are satisfied and therefore φ is satisfied by the truth assignment, implying that φ is satisfiable.

In the other direction, we have to argue that if φ is satisfiable, then G can be 3-colored. Fix a satisfying assignment of G. Color vertices t, f, and b with colors red, green, and blue in that order. Red signifies true and green

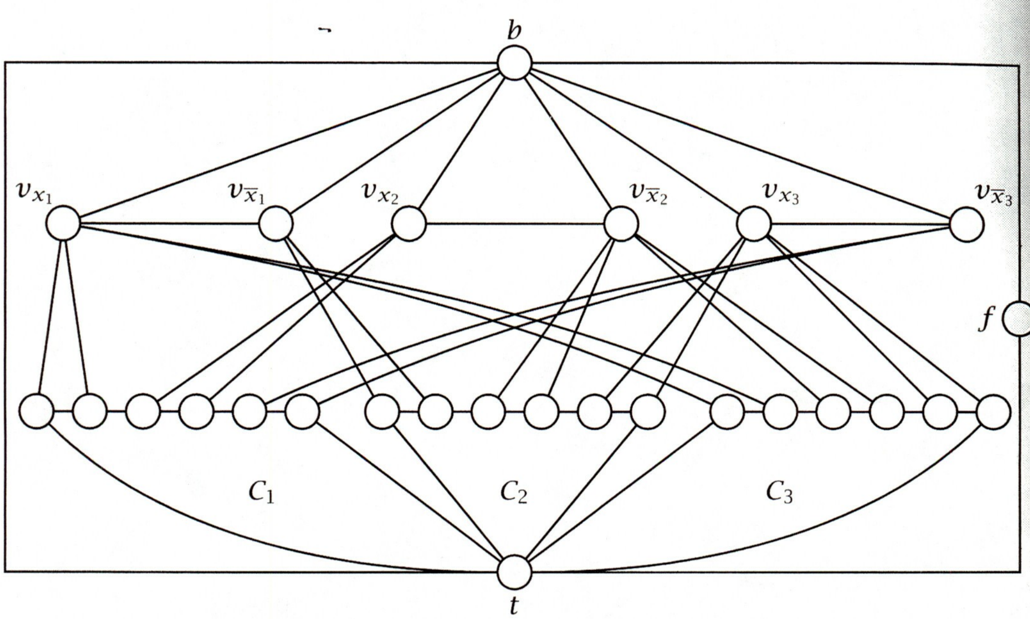

Figure 10.4.7 The graph G for the formula $\varphi = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee x_3)$. On top are the three triangles for x_1, x_2, x_3; below are the three clause gadgets, one for each clause. For example, the top of the middle triangle of the C_2 gadget is $v_{\overline{x}_2}$ because the second clause contains the literal $\overline{x}_2$. Finally, the triangle $\{b, f, t\}$ surrounds the rest of the graph, the edge b to t on the left side and edges t to f and f to b on the right side.

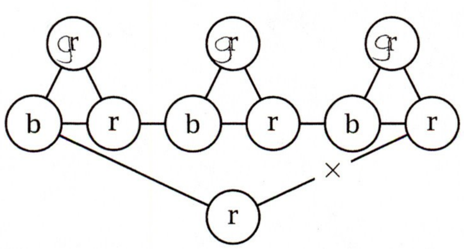

Figure 10.4.8 Color conflict in the clause gadget: If all tops are green, then the colors in the baseline have to alternate between red and blue. Consequently, one of the endpoints of the baseline is red, conflicting with the red t.

signifies false. For each variable x in φ, color vertex v_x red and $v_{\overline{x}}$ green, if x is true, and v_x green and $v_{\overline{x}}$ red, if x is false. Finally, we have to show how to color the gadgets. Since we chose a satisfying assignment, at least one of the three top vertices of the triangles has to be red. Figure 10.4.9 shows how to color the gadget in each of these cases. For example, if all of the tops are red, we color the nodes of the baseline alternately with colors green and blue.

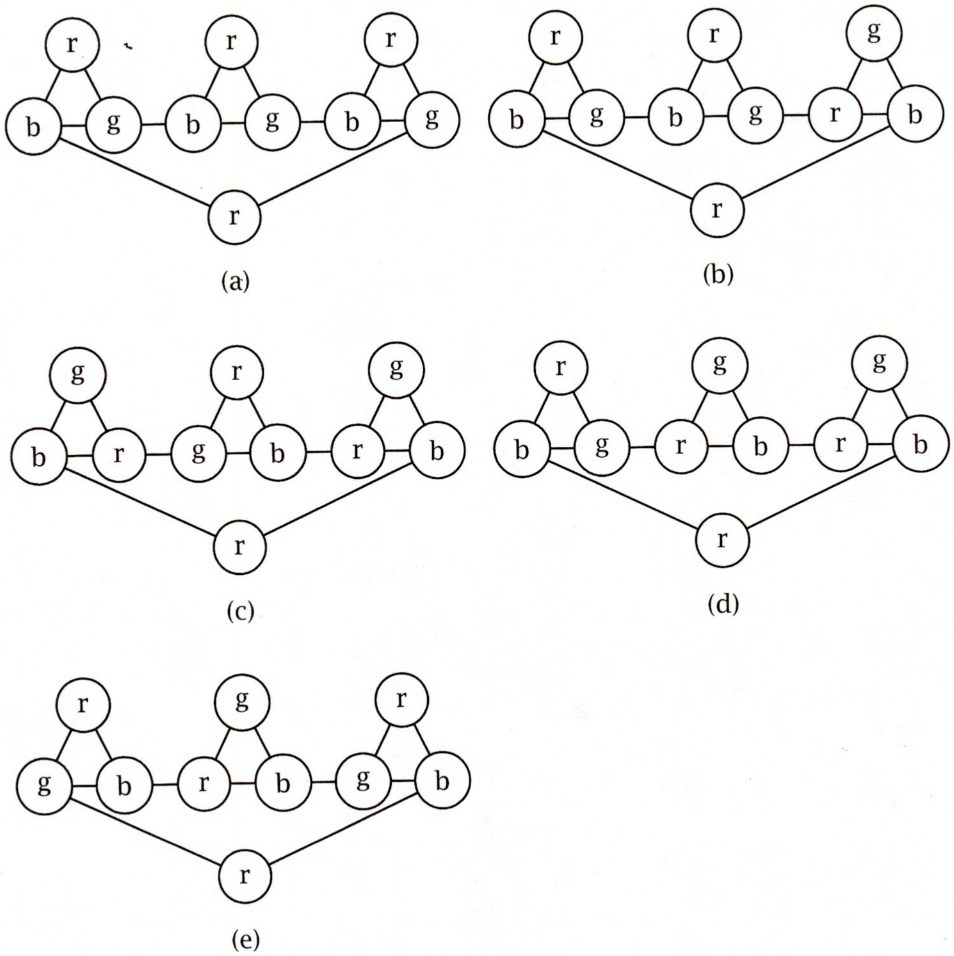

Figure 10.4.9 How to color the clause gadgets: when all tops are red (a), when one of the outer tops is green (b), when both of the outer tops are green (c), when one of the outer tops is red (d), when both of the outer tops are red (e).

We have shown that G is 3-colorable if and only if φ is satisfiable. This establishes that satisfiability reduces to graph 3-colorability, which, therefore, is **NP**-complete. ∎

The result can be strengthened to show that deciding whether a planar graph is 3-colorable is **NP**-complete, although we know that all planar graphs are 4-colorable by the four-color theorem. From the **NP**-completeness of the planar case, it also follows that CROSSWORD is **NP**-complete, since we can make the construction of Example 10.4.1 work for general planar graphs.