columns:

$$A = \left[\begin{array}{c|ccccccccccc} & -1 & -1 & \dots & -1 & 0 & \dots & 0 & 0 & \dots & 0 \\ & 0 & 0 & & 0 & -1 & \dots & -1 & \vdots & & 0 \\ B & \vdots & & & \vdots & & & \ddots & 0 & \dots & 0 \\ & 0 & 0 & \dots & 0 & 0 & \dots & 0 & -1 & \dots & -1 \end{array}\right]$$

We show that the instance of 3-SAT is a "Yes" instance if and only if this instance of 0-1 INTEGER PROGRAMMING is a "Yes" instance, that is, if and only if $A\boldsymbol{x} = \boldsymbol{d}$.

We write the column p-vector $\boldsymbol{x}$ as the concatenation of the column m-vector $\boldsymbol{u}$ and the column mn-vector $\boldsymbol{v}$. It follows that $A\boldsymbol{x} = \boldsymbol{b}$ if and only if $A\boldsymbol{u} \geq \boldsymbol{b}$. Now consider the ith component of $A\boldsymbol{u}$. Let $\boldsymbol{u}$ select k_i uncomplemented and l_i complemented variables of clause c_i. Then, $A\boldsymbol{u} \geq \boldsymbol{b}$ if and only if $k_i - l_i \geq d_i = 1 - q_i$ or $k_i + (q_i - l_i) \geq 1$ for all i. Now let $x_i = u_i$ for $1 \leq i \leq n$. Then k_i and $q_i - l_i$ are the numbers of uncomplemented and complemented variables in c_i that are set to 1 and 0, respectively. Since $k_i + (q_i - l_i) \geq 1$, c_i is satisfied, as are all clauses, giving us the desired result. ■

8.11 The Boundary Between **P** and **NP**

It is important to understand where the boundary lies between problems in **P** and the **NP**-complete problems. While this topic is wide open, we shed a modest amount of light on it by showing that 2-SAT, the version of 3-SAT in which each clause has at most two literals, lies on the **P**-side of this boundary, as shown below. In fact, it is in **NL**, which is in **P**.

THEOREM 8.11.1 2-SAT *is in* **NL**.

Proof Given an instance I of 2-SAT, we first insure that each clause has exactly two distinct literals by adding to each one-literal clause a new literal z that is not used elsewhere. We then construct a directed graph $G = (V, E)$ with vertices V labeled by the literals x and $\overline{x}$ for each variable x appearing in I. This graph has an edge (α, β) in E directed from vertex α to vertex β if the clause $(\overline{\alpha} \vee \beta)$ is in I. If $(\overline{\alpha} \vee \beta)$ is in I, so is $(\beta \vee \overline{\alpha})$ because of commutativity of $\vee$. Thus, if $(\alpha, \beta) \in E$, then $(\overline{\beta}, \overline{\alpha}) \in E$ also. (See Fig. 8.15.) Note that $(\alpha, \beta) \neq (\overline{\beta}, \overline{\alpha})$ because this requires that $\beta = \overline{\alpha}$, which is not allowed. Let $\alpha \neq \overline{\gamma}$. It follows that if there is a path from α to γ in G, there is a distinct path from $\overline{\gamma}$ to $\overline{\alpha}$ obtained by reversing the directions of each edge on the path and replacing the literals by their complements.

To understand why these edges are chosen, note that if all clauses of I are satisfied and $(\overline{\alpha} \vee \beta)$ is in I, then $\alpha = 1$ implies that $\beta = 1$. This implication relation, denoted $\alpha \Rightarrow \beta$, is transitive. If there is a path $(\alpha_1, \alpha_2, \dots, \alpha_k)$ in G, then there are clauses $(\overline{\alpha}_1 \vee \alpha_2)$, $(\overline{\alpha}_2 \vee \alpha_3), \dots, (\overline{\alpha}_{k-1} \vee \alpha_k)$ in I. If all clauses are satisfied and if the literal $\alpha_1 = 1$, then each un-negated literal on this path must have value 1.

We now show that an instance I is a "No" instance if and only if there is a variable x such that there is a path in G from x to $\overline{x}$ and one from $\overline{x}$ to x.

If there is a variable x such that such paths exists, this means that $x \Rightarrow \overline{x}$ and $\overline{x} \Rightarrow x$ which is a logical contradiction. This implies that the instance I is a "No" instance.

Conversely, suppose I is a "No" instance. To prove there is a variable x such that there are paths from vertex x to vertex $\overline{x}$ and from $\overline{x}$ to x, assume that for no variable x does this

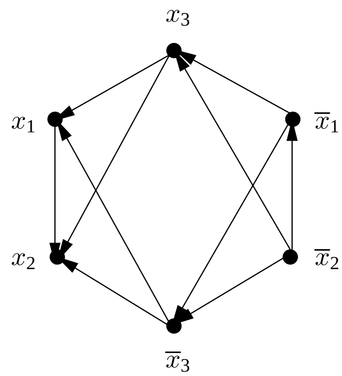

Figure 8.15 A graph capturing the implications associated with the following satisfiable instance of 2-SAT: $(x_3 \vee x_2) \wedge (x_3 \vee x_1) \wedge (\overline{x}_3 \vee x_2) \wedge (\overline{x}_1 \vee x_2) \wedge (\overline{x}_3 \vee x_1)$.

condition hold and show that I is a "Yes" instance, that is, every clause is satisfied, which contradicts the assumption that I is a "No" instance.

Identify a variable that has not been assigned a value and let α be one of the two corresponding literals such that there is no directed path in G from the vertex α to $\overline{\alpha}$. (By assumption, this must hold for at least one of the two literals associated with x.) Assign value 1 to α and each literal λ reachable from it. (This assigns values to the variables identified by these literals.) If these assignments can be made without assigning a variable both values 0 and 1, each clause can be satisfied and I is "Yes" instance rather than a "No" one, as assumed. To show that each variable is assigned a single value, we assume the converse and show that the conditions under which values are assigned to variables by this procedure are contradicted. A variable can be assigned contradictory values in two ways: a) on the current step the literals λ and $\overline{\lambda}$ are both reachable from α and assigned value 1, and b) a literal λ is reachable from α on the current step that was assigned value 0 on a previous step. For the first case to happen, there must be a path from α to vertices λ and $\overline{\lambda}$. By design of the graph, if there is a path from α to $\overline{\lambda}$, there is a path from λ to $\overline{\alpha}$. Since there is a path from α to λ, there must be a path from α to $\overline{\alpha}$, contradicting the assumption that there are no such paths. In the second case, let a λ be assigned 1 on the current step that was assigned 0 on a previous step. It follows that $\overline{\lambda}$ was given value 1 on that step. Because there is a path from α to λ, there is one from $\overline{\lambda}$ to $\overline{\alpha}$ and our procedure, which assigned $\overline{\lambda}$ value 1 on the earlier step, must have assigned $\overline{\alpha}$ value 1 on that step also. Thus, α had the value 0 before the current step, contradicting the assumption that it was not assigned a value.

To show that 2-SAT is in **NL**, recall that **NL** is closed under complements. Thus, it suffices to show that "No" instances of 2-SAT can be accepted in nondeterministic logarithmic space. By the above argument, if I is a "No" instance, there is a variable x such that there is a path in G from x to $\overline{x}$ and from $\overline{x}$ to x. Since the number of vertices in G is at most linear in n, the length of I (it may be as small as $O(\sqrt{n})$), an NDTM can propose and then verify in space $O(\log n)$ a path in G from x to $\overline{x}$ and back by checking that the putative edges are edges of G, that x is the first and last vertex on the path, and that $\overline{x}$ is encountered before the end of the path. ■