several Hamiltonian cycles; one visits the nodes in the order 1, 6, 4, 3, 2, 5, 1, while another visits the nodes in the order 1, 2, 4, 5, 6, 3, 1.

The Hamiltonian Cycle Problem is then simply the following:

Given a directed graph G, does it contain a Hamiltonian cycle?

Proving *Hamiltonian Cycle* is NP-Complete

We now show that both these problems are NP-complete. We do this by first establishing the NP-completeness of Hamiltonian Cycle, and then proceeding to reduce from Hamiltonian Cycle to Traveling Salesman.

(8.17) Hamiltonian Cycle *is NP-complete.*

Proof. We first show that Hamiltonian Cycle is in $\mathcal{NP}$. Given a directed graph $G = (V, E)$, a certificate that there is a solution would be the ordered list of the vertices on a Hamiltonian cycle. We could then check, in polynomial time, that this list of vertices does contain each vertex exactly once, and that each consecutive pair in the ordering is joined by an edge; this would establish that the ordering defines a Hamiltonian cycle.

We now show that 3-SAT $\leq_P$ Hamiltonian Cycle. Why are we reducing from 3-SAT? Essentially, faced with Hamiltonian Cycle, we really have no idea *what* to reduce from; it's sufficiently different from all the problems we've seen so far that there's no real basis for choosing. In such a situation, one strategy is to go back to 3-SAT, since its combinatorial structure is very basic. Of course, this strategy guarantees at least a certain level of complexity in the reduction, since we need to encode variables and clauses in the language of graphs.

So consider an arbitrary instance of 3-SAT, with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_k$. We must show how to solve it, given the ability to detect Hamiltonian cycles in directed graphs. As always, it helps to focus on the essential ingredients of 3-SAT: We can set the values of the variables however we want, and we are given three chances to satisfy each clause.

We begin by describing a graph that contains 2^n different Hamiltonian cycles that correspond very naturally to the 2^n possible truth assignments to the variables. After this, we will add nodes to model the constraints imposed by the clauses.

We construct n paths $P_1, \ldots, P_n$, where P_i consists of nodes $v_{i1}, v_{i2}, \ldots, v_{ib}$ for a quantity b that we take to be somewhat larger than the number of clauses k; say, $b = 3k + 3$. There are edges from v_{ij} to $v_{i,j+1}$ and in the other direction from $v_{i,j+1}$ to v_{ij}. Thus P_i can be traversed "left to right," from v_{i1} to v_{ib}, or "right to left," from v_{ib} to v_{i1}.

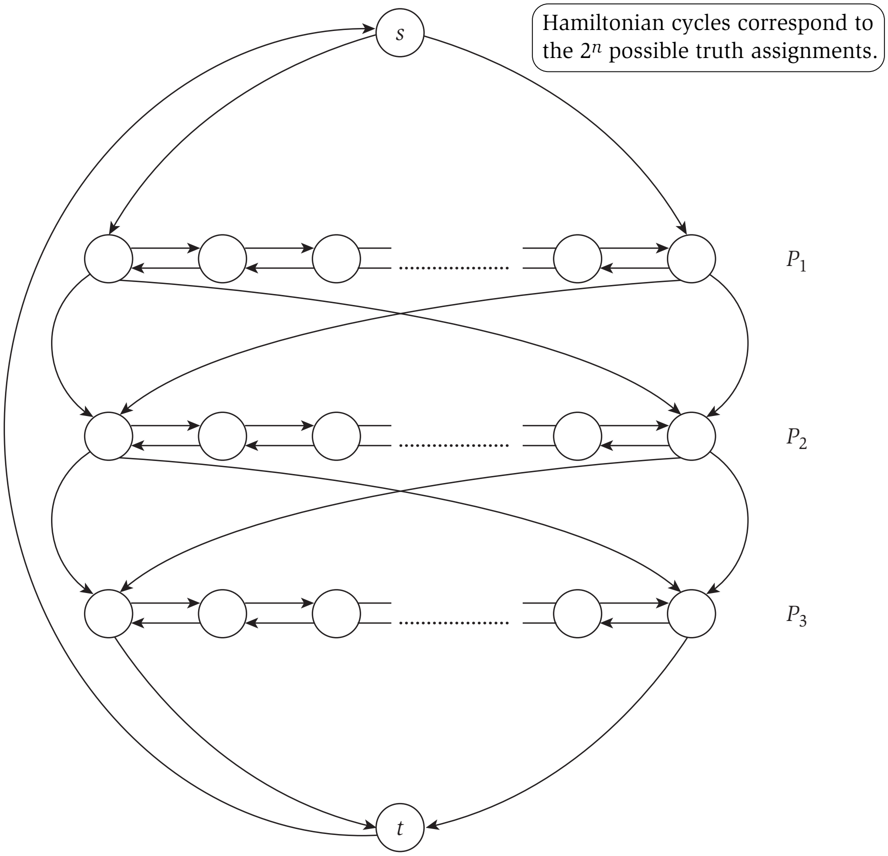

Figure 8.7 The reduction from 3-SAT to Hamiltonian Cycle: part 1.

We hook these paths together as follows. For each $i = 1, 2, \ldots, n-1$, we define edges from v_{i1} to $v_{i+1,1}$ and to $v_{i+1,b}$. We also define edges from v_{ib} to $v_{i+1,1}$ and to $v_{i+1,b}$. We add two extra nodes s and t; we define edges from s to v_{11} and v_{1b}; from v_{n1} and v_{nb} to t; and from t to s.

The construction up to this point is pictured in Figure 8.7. It's important to pause here and consider what the Hamiltonian cycles in our graph look like. Since only one edge leaves t, we know that any Hamiltonian cycle $\mathcal{C}$ must use the edge (t, s). After entering s, the cycle $\mathcal{C}$ can then traverse P_1 either left to right or right to left; regardless of what it does here, it can then traverse P_2 either left to right or right to left; and so forth, until it finishes traversing P_n and enters t. In other words, there are exactly 2^n different Hamiltonian cycles, and they correspond to the n independent choices of how to traverse each P_i.

This naturally models the n independent choices of how to set each variables $x_1, \ldots, x_n$ in the 3-SAT instance. Thus we will identify each Hamiltonian cycle uniquely with a truth assignment as follows: If $\mathcal{C}$ traverses P_i left to right, then x_i is set to 1; otherwise, x_i is set to 0.

Now we add nodes to model the clauses; the 3-SAT instance will turn out to be satisfiable if and only if any Hamiltonian cycle survives. Let's consider, as a concrete example, a clause

$$C_1 = x_1 \vee \overline{x_2} \vee x_3.$$

In the language of Hamiltonian cycles, this clause says, "The cycle should traverse P_1 left to right; or it should traverse P_2 right to left; or it should traverse P_3 left to right." So we add a node c_1, as in Figure 8.8, that does just this. (Note that certain edges have been eliminated from this drawing, for the sake of clarity.) For some value of ℓ, node c_1 will have edges *from* $v_{1\ell}$, $v_{2,\ell+1}$, and $v_{3\ell}$; it will have edges *to* $v_{1,\ell+1}$, $v_{2,\ell}$, and $v_{3,\ell+1}$. Thus it can be easily spliced into any Hamiltonian cycle that traverses P_1 left to right by visiting node c_1 between $v_{1\ell}$ and $v_{1,\ell+1}$; similarly, c_1 can be spliced into any Hamiltonian cycle that traverses P_2 right to left, or P_3 left to right. It cannot be spliced into a Hamiltonian cycle that does not do any of these things.

More generally, we will define a node c_j for each clause C_j. We will reserve node positions $3j$ and $3j+1$ in each path P_i for variables that participate in clause C_j. Suppose clause C_j contains a term t. Then if $t = x_i$, we will add edges $(v_{i,3j}, c_j)$ and $(c_j, v_{i,3j+1})$; if $t = \overline{x_i}$, we will add edges $(v_{i,3j+1}, c_j)$ and $(c_j, v_{i,3j})$.

This completes the construction of the graph G. Now, following our generic outline for NP-completeness proofs, we claim that the 3-SAT instance is satisfiable if and only if G has a Hamiltonian cycle.

First suppose there is a satisfying assignment for the 3-SAT instance. Then we define a Hamiltonian cycle following our informal plan above. If x_i is assigned 1 in the satisfying assignment, then we traverse the path P_i left to right; otherwise we traverse P_i right to left. For each clause C_j, since it is satisfied by the assignment, there will be at least one path P_i in which we will be going in the "correct" direction relative to the node c_j, and we can splice it into the tour there via edges incident on $v_{i,3j}$ and $v_{i,3j+1}$.

Conversely, suppose that there is a Hamiltonian cycle $\mathcal{C}$ in G. The crucial thing to observe is the following. If $\mathcal{C}$ enters a node c_j on an edge from $v_{i,3j}$, it must depart on an edge to $v_{i,3j+1}$. For if not, then $v_{i,3j+1}$ will have only one unvisited neighbor left, namely, $v_{i,3j+2}$, and so the tour will not be able to visit this node and still maintain the Hamiltonian property. Symmetrically, if it enters from $v_{i,3j+1}$, it must depart immediately to $v_{i,3j}$. Thus, for each node c_j,

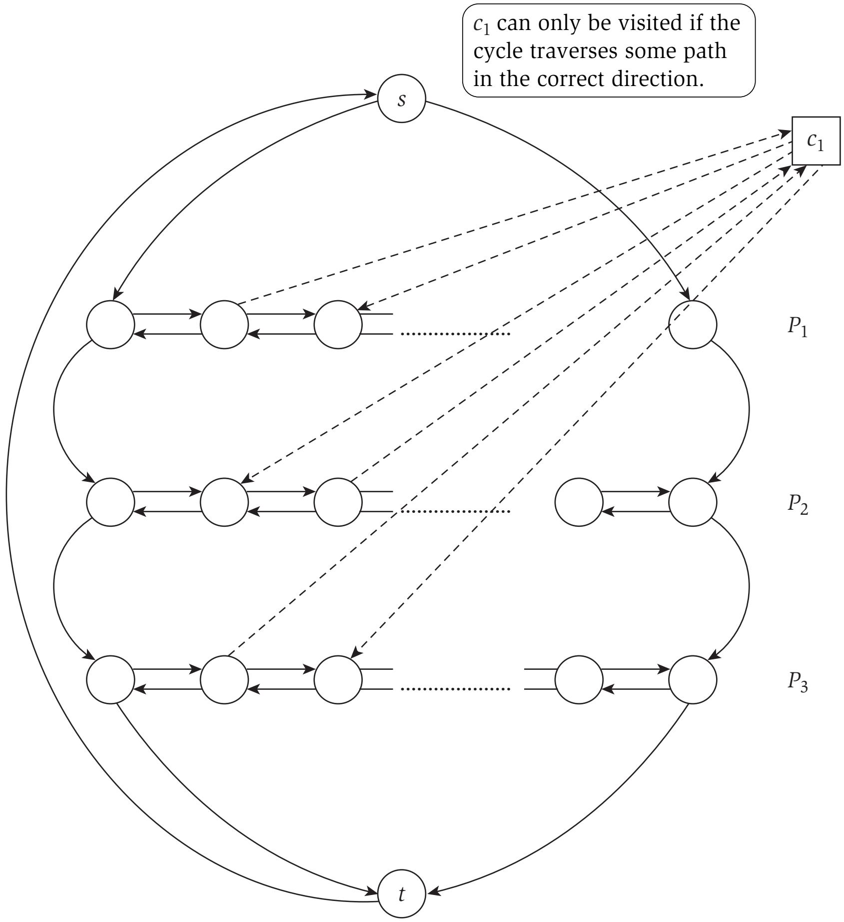

Figure 8.8 The reduction from 3-SAT to Hamiltonian Cycle: part 2.

the nodes immediately before and after c_j in the cycle $\mathcal{C}$ are joined by an edge e in G; thus, if we remove c_j from the cycle and insert this edge e for each j, then we obtain a Hamiltonian cycle $\mathcal{C}'$ on the subgraph $G - \{c_1, \ldots, c_k\}$. This is our original subgraph, before we added the clause nodes; as we noted above, any Hamiltonian cycle in this subgraph must traverse each P_i fully in one direction or the other. We thus use $\mathcal{C}'$ to define the following truth assignment for the 3-SAT instance. If $\mathcal{C}'$ traverses P_i left to right, then we set $x_i = 1$; otherwise we set $x_i = 0$. Since the larger cycle $\mathcal{C}$ was able to visit each clause node c_j, at least one of the paths was traversed in the "correct" direction relative to the node c_j, and so the assignment we have defined satisfies all the clauses.