Variations on the Hamiltonian Cycle Problem

Karthik Gopalan

September 29, 2014

1 Abstract

The Hamiltonian Cycle Problem is easily stated as: given a graph G, with verticies V and edges E, find a *cycle* that visits every vertex. Everyone has heard that it is NP-Complete but most people have not seen the proof. This paper will enlighten those people by providing detailed proofs of why 4 classic variations of the problem are NP-Complete.

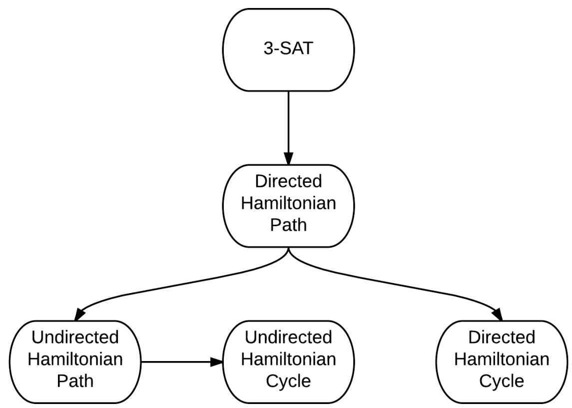
2 Introdution

The Hamiltonian Cycle Problem is easily stated as: given a graph G, with verticies V and edges E, find a *cycle* that visits every vertex. A natural variation would be given a graph G, find a *path* that starts at some vertex $s \in V$, visits each vertex exactly once, and ends at vertex $t \in V$. Other problems can be synthesized by varying the graph G. This paper will focus on proving that the following variations are NP-Complete

1. {G | G is a directed graph with a directed Ham Cycle}
2. {G | G is a directed graph with a directed Ham Path}
3. {G | G is an undirected graph with an undirected Ham Cycle}
4. {G | G is an undirected graph with an undirected Ham Path}

To prove that all of these problems are NP-Complete, we first show that 3-SAT, a known NP-complete problem, is polynomial time reducible to problem 2. Then we show that problem 2 is polynomial time reducible to problems 1 and 4, and finally that problem 4 is poly time reducible to problem 3.

Here is a flowchart of the proof:



All of the credit for my understanding of these proofs goes to Michael Sipster and his book, *Introduction to the Theory of Computation*.

3 3-SAT $\leq_P$ Directed Ham Path

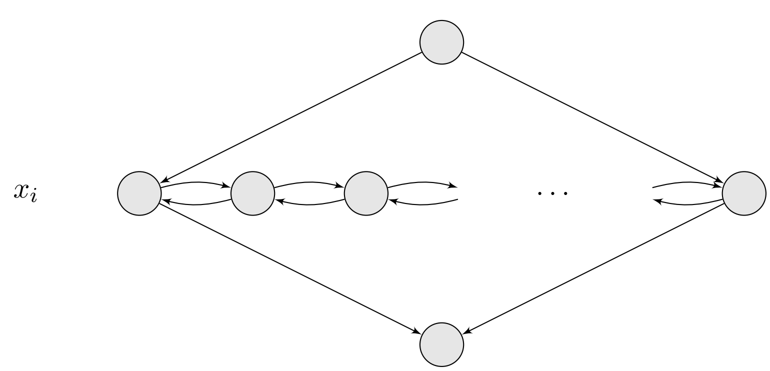
The Hamiltonian Path problem is in NP because a solution to the problem can be verified in polynomial time with a certificate. The certificate is the path represented by an ordering of nodes. To verify a Hamiltonian path; make sure each node is in the path once, and for each consecutive pair of nodes in the list verify that an edge exists between them. This can obviously be done in polynomial time.

The boolean satisfiability problem (SAT) is: given a boolean formula ϕ with n variables seperated by $\wedge$'s $\vee$'s and $\neg$'s, find an assignment of the variables such that $\phi = 1$. 3-SAT is an instance of SAT where the formula is in conjunctive normal form with clauses of size 3. 3-SAT is NP-Complete.

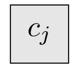
Given a 3cnf-formula ϕ with k clauses, n variables $x_1 \cdots x_n$, and literals x_i, $\overline{x_i}$ represented by a, b, or c.

$$\phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \cdots \wedge (a_k \vee b_k \vee c_k)$$

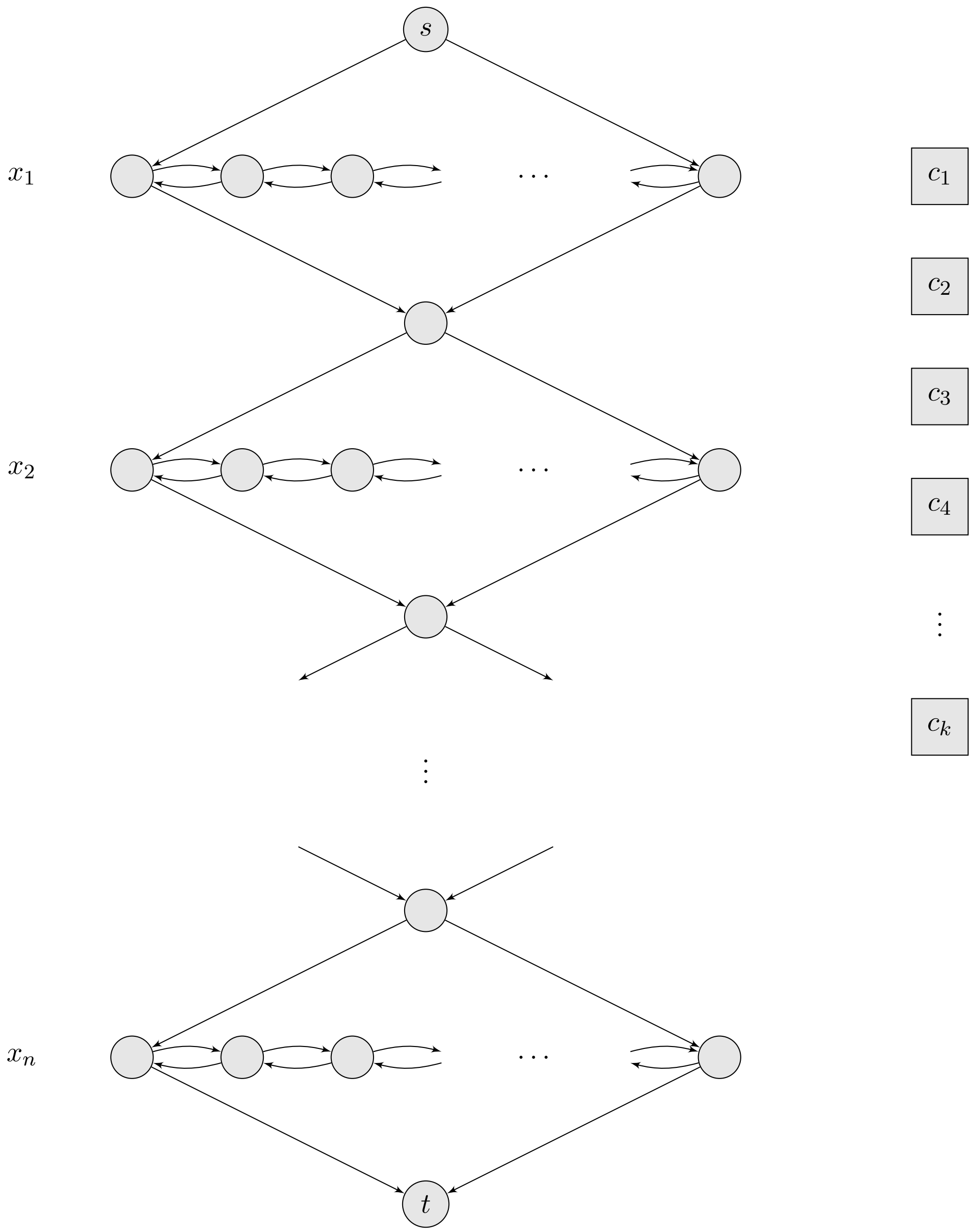
We must turn ϕ into a directed graph G that holds the property that a Hamiltonian Path between verticies s and t exists if and only if ϕ has a satisfing assignment. To start, we must represent each variable x_i with a gadget that contains a horizontal row of $3k+1$ nodes, a node with edges to each end of the row, and a node with incoming edges from each end of the row. The structure looks like this:



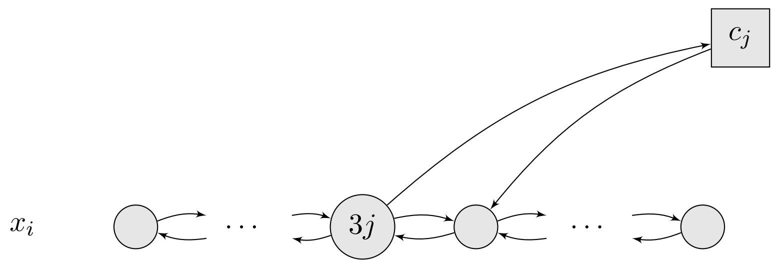
Every clause is represented as a single node like so:



When all of the variables are connected the graph will look like:



All that is left is to insert the edges connecting the clauses. If the literal x_i appears in clause c_j we will go to the horizontal row of x_i. Then add an edge from the $3j^{th}$ node to node c_j and an edge from node c_j to the $3j+1^{th}$ node. If the literal $\overline{x_i}$ appears in clause c_j we will go to the horizontal row of x_i. Then add an edge from the $3j+1^{th}$ node to node c_j and an edge from node c_j to the $3j^{th}$ node. The first case is illustrated below.



Now that the clauses are connected, the graph is complete but how do we know it works?

3.1 Proof

There are only two ways for a path to touch every node in each diamond shaped gadget.

1. Start at the top, follow the left edge, traverse the horizontal row to the *right*, and finally take the edge to the bottom node.
2. Start at the top, follow the right edge, traverse the horizontal row to the *left*, and take the edge to the bottom node.

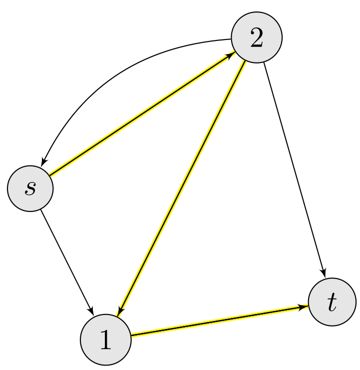
Now assume that ϕ is satisfiable. If x_i is set to true, follow path 1 through the appropriate diamond. If x_i is set to false, follow path 2. In order to complete the path, we must connect the clause nodes. For each clause, pick a literal that was assigned true. If we pick x_i in clause c_j redirect our path from the $3j^{th}$ node to the clause node and back to the $3j+1^{th}$ node. If $\overline{x_i}$ is assigned true in clause c_j, then x_i is false and we are following path 2 through the diamond. Therefore, the path is going the right direction to allow a detour from the $3j+1^{th}$ node to the clause node and a return to the $3j^{th}$ node. Now that the clause nodes are included, the Hamiltonian Path is complete.

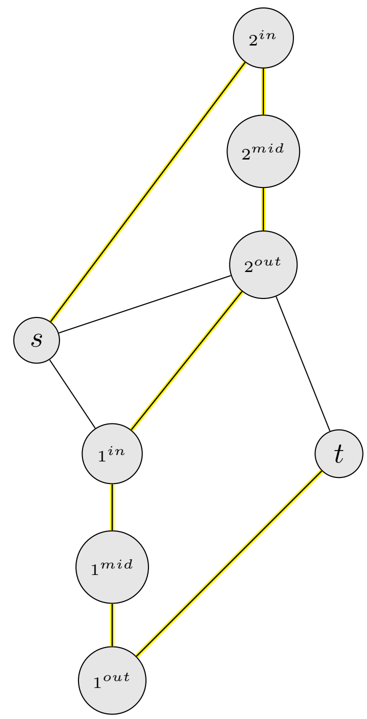
To finish the proof we must show that if G has a Hamiltonian Path, ϕ has a satisfying assignment. In order for the satisfying assignment to be read from the graph, every horizontal row of nodes must be traversed from left to right or right to left. If they are mixed, the variable's assignment will be unclear. The only way to get a mixed assignment is if a path enters a clause from one diamond and then follows an edge to another. If we eventually got back to the original diamond, the only way to visit every node in the row is to traverse the opposite way. Therefore, the path will be stuck inside of a diamond and will not be able to touch every vertex. So any Hamiltonian Path must start at the top of the structure, traverse each horizontal row in one direction, and end at the bottom. This path guaruntees a satisfying assignment of ϕ. Finally, the reduction takes time $O(|n(3k+1)+k|)$ which is polynomial.

4 Directed Ham Path $\leq_P$ Undirected Ham Path

To show that the Undirected Hamiltonian Path problem is NP-Complete, we will show a polynomial time reduction from the directed version of the problem.

Start with a directed graph G with a Hamiltonian Path from s to t and construct an undirected graph G' with a Hamiltonian Path from s^{out} to t^{in}. In order to simulate a directed graph, we will replace each node $u \in E - s, t$ with three nodes; u^{in}, u^{mid}, and u^{out}. For each triplet, add edges connecting u^{in} and u^{out} to u^{mid}. Finally, add an edge from a^{out} to b^{in} if there exists an edge from a to b in G. This tranformation is illustrated below.





I claim that G has a Hamiltonian Path from s to t if and only if G' has a Hamiltonian Path from s' to t'. To prove the left to right implication, imagine a Hamiltonian path starts at s, passes through some ordering of u_i's, and finally ends at t. Like so:

$$s \to u_1 \to u_2 \to u_3 \to \ldots \to u_{|E|-2} \to t$$

Since our construction just replaces nodes by triplets of nodes, the same path in G' would look like:

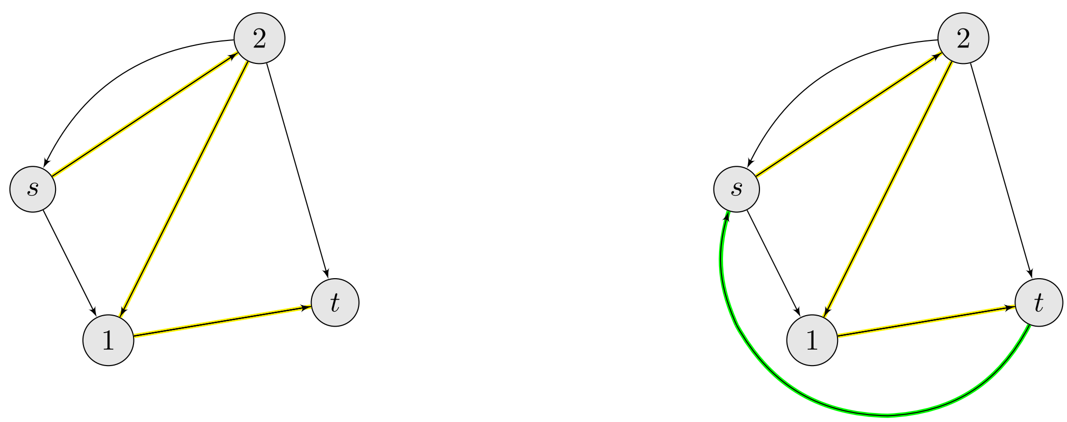
$$s_{out} \to u_1^{in} \to u_1^{mid} \to u_1^{out} \to u_2^{in} \to u_2^{mid} \to u_2^{out} \to \ldots \to t^{in}$$

To prove the right to left implication, I will describe the Hamiltonian Path in G'. Start at s^{out} and follow an edge to some u_i^{in}. We have to go to u_i^{mid} because there is no other way to incorporate it in the path. The next node has to be u_i^{out} and from there we can take an edge to u_j^{in} and traverse another triplet. If you repeat this procedure until you reach t^{in} you will derive a path that resembles the one above. Since paths of this form have a directed equivalent, the proof is complete. This reduction takes time $O(2|V| + 2|E|)$ since you need to add two verticies and two nodes in G' for almost every node in G.

5 Dir Ham Path $\leq_P$ Dir Ham Cycle AND Undir Ham Path $\leq_P$ Undir Ham

I know what you are thinking: "Karthik how are you going to show two poly time reductions in one section!??" I'll show you.

Given a graph $G = (V, E)$ (directed or undirected) with a Hamiltonian Path from s to t , I will construct the graph $G' = (V, E \cup (t, s))$ by just adding an edge from t to s. Here is an illustration of this construction on a simple directed graph.



I claim that G has a Hamiltonian Path from s to t if and only if G' has a Hamiltonian Cycle. Lets say G has a Hamiltonian Path, if you follow the path from s to t in G' then take the edge from t to s you will arrive back at s, thus completeing the cycle. If G' has a Hamiltonian Cycle; start at s, follow the cycle to t, and then stop. This is your Hamiltonian Path.

This procedure takes constant time with respect to the size of the input.