Question: Show that SAT can be polynomially reduced to integer linear programming problem. Hence prove that Integer Linear Program is NP-complete.

Integer linear programming (ILP) is like linear programming, with the additional constraint that all variables must take on integral values. The decision version of integer programming asks if there exists a point satisfying all the constraints (for the decision version there is no objective function).

To claim ILP is NP-complete, we must show the following:

- ILP is in NP
- We can reduce 3SAT to ILP

Let the variables in the 3SAT formula be x_1, x_2, ..., x_n. We will have corresponding variables z_1, z_2, ..., z_n in our integer linear program. First, we restrict each variable to be 0 or 1:

$$0 \leq z_i \leq 1 \qquad \forall\, i$$

Assigning $z_i = 1$ in the integer program represents setting $x_i = true$ in the formula, and assigning $z_i = 0$ represents setting $x_i = false$.

For each clause like $(x_1 \quad OR \quad NOT(x_2) \quad OR \quad x_3)$ have a constraint like:

$$z_1 + (1 - z_2) + z_3 > 0$$

To satisfy this inequality we must either set $z_1 = 1$ or $z_2 = 0$ or $z_3 = 1$, which means we either set $x_1 = true$ or $x_2 = false$ or $x_3 = true$ in the corresponding truth assignment.

Question:

Pick an NP-complete problem of your choice, S, polynomially reduce S to Hamiltonian Circuit problem to prove that HCP is NP-complete.

Given a directed graph G, if there exists a cycle that visits every vertex exactly once, such cycle is called a Hamiltonian cycle.

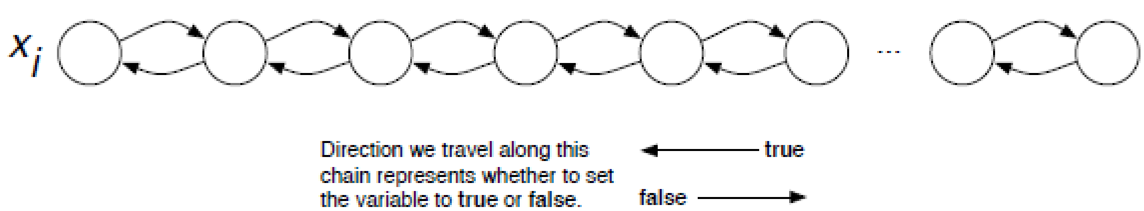
To prove that a Hamiltonian circuit is NP-complete:

- First, we show the Hamiltonian Circuit belong to NP.
- Second, we show that 3-SAT can be polynomially reduced to Hamiltonian Circuit.
- In other words, how do we encode an instance I of 3-SAT as a graph G such that I is satisfiable exactly when G has a Hamiltonian cycle.

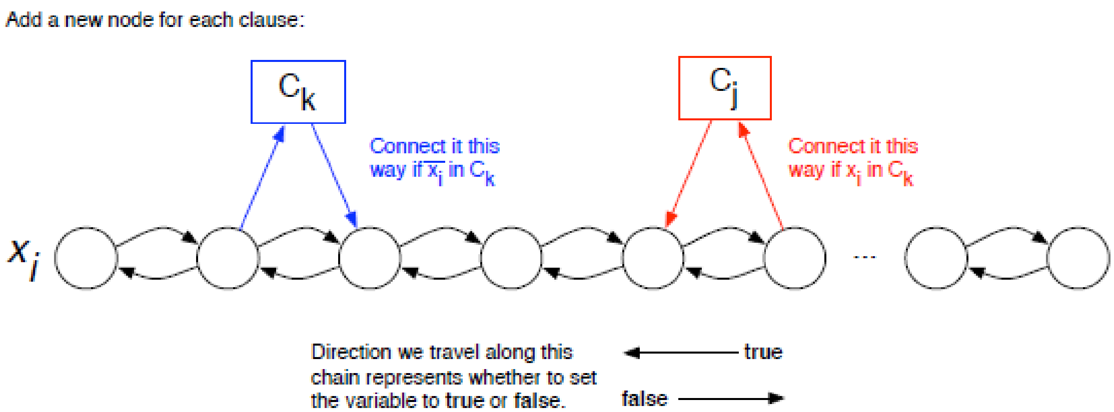
Consider an instance I of 3-SAT, with variables x_1,…..,x_n and clauses C_1,…..,C_k.

- Create some graph structure (a 'gadget') that represents the variables as shown below.
- Add some graph structure that represents the clauses.
- Hook them up in some way that encodes the formula.
- Show that this graph has a Hamiltonian Circuit if and only if the formula is satisfiable.

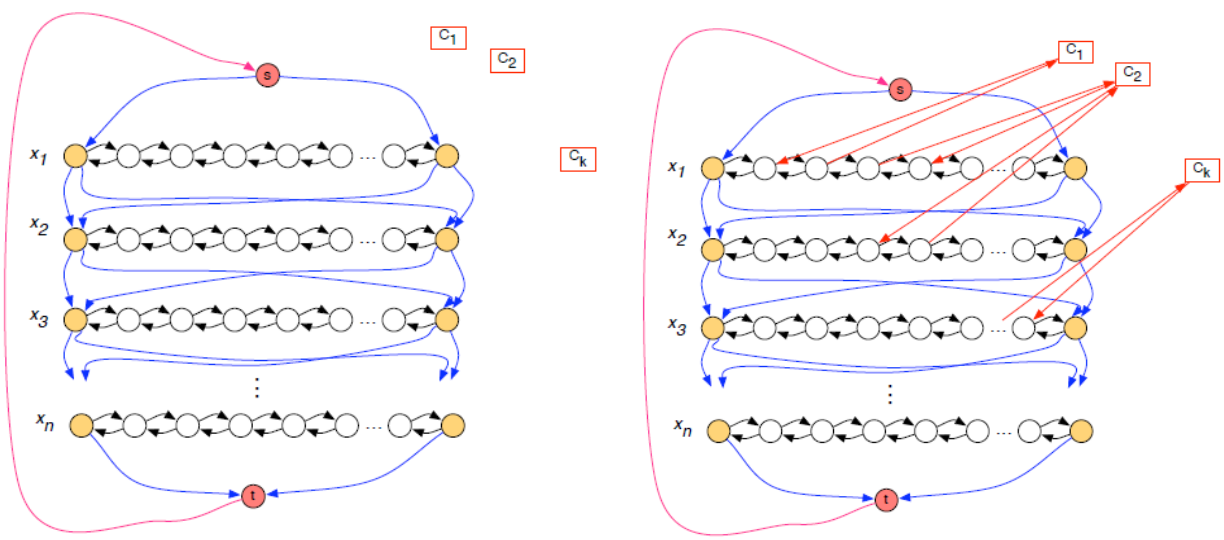
Gadget Representing the Variables:



Hooking in the Clauses:



Connecting the paths:



Therefore, a Hamiltonian path encodes a truth assignment for the variables (depending on which direction each chain is traversed. For there to be a Hamiltonian cycle, we must visit every clause node, but we can only visit a clause if we satisfy it by setting one of its terms to TRUE.

Hence, if there is a Hamiltonian cycle, there is a satisfying assignment, proving Hamiltonian Circuit to be NP-complete.