Preliminaries

The reduction technique revisited: a way to show that a problem is **NP**-complete. Assume a problem (a language) PROB that we wish to show to be **NP**-complete. First, we have to show that PROB is in **NP**, i.e., that there is a non-deterministic polynomial Turing machine deciding whether an arbitrary input string belongs to PROB. Second, we show that PROB is **NP**-hard by the following technique: We take a problem that is known to be **NP**-complete, such as SAT or CLIQUE, and reduce it to PROB. The reduction is a log-space computable function that maps each instance of the chosen **NP**-complete problem into an instance of PROB in a way that the original instance is a "yes"-instance of the chosen **NP**-complete problem iff the mapped instance is a "yes"-instance of PROB.

A Proof that LONGEST PATH is NP-complete

The problem LONGEST PATH is: Given an undirected graph $G = \langle V, E \rangle$ and a positive (binary coded) integer $K \leq |V|$, does G have a simple path (that is, a path encountering no vertex more than once) with K or more edges?

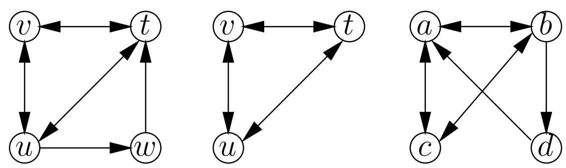
We use a simple reduction from HAMILTON PATH problem: Given an undirected graph, does it have a Hamiton path, i.e., a path visiting each vertex exactly once? HAMILTON PATH is **NP**-comple (book, page 193). Given an instance $G' = \langle V', E' \rangle$ for HAMILTON PATH, count the number $|V'|$ of nodes in G' and output the instance $G = G', K = |V'|$ for LONGEST PATH. Obviously, G' has a simple path of length $|V'|$ iff G' has a Hamilton path.

Furthermore, consider the following variant of LONGEST PATH: Given an undirected graph $G = \langle V, E \rangle$, two vertices $v, v' \in V$, and a positive (binary coded) integer $K \leq |V|$, does G have a simple path (that is, a path encountering no vertex more than once) with K or more edges from v to v'? We can use the same reduction from HAMILTON PATH BETWEEN TWO VERTICES problem: Given an undirected graph and two of its vertices, does it have a Hamiton path between the given vertices? It is known that HAMILTON PATH BETWEEN TWO VERTICES is **NP**-complete, see, e.g., Garey and Johnson: "Computers and Intractability – A Guide to the Theory of **NP**-Completeness".

A Proof that SUBGRAPH ISOMORPHISM is NP-complete

Preliminaries

A graph G is a pair $\langle V, E\rangle$ such that V is a finite set of vertices and $E \subseteq V \times V$ is the set of edges between vertices. A subgraph G' of G is a graph $\langle V', E'\rangle$ such that $V' \subseteq V$ and $E' \subseteq E \cap V' \times V'$. Two graphs, $G_1 = \langle V_1, E_1\rangle$ and $G_2 = \langle V_2, E_2\rangle$, are isomorphic iff there exists a bijective mapping $f : V_1 \to V_2$ such that $\langle v_1, v_2\rangle \in E_1 \Leftrightarrow \langle f(v_1), f(v_2)\rangle \in E_2$. As an example, consider the graphs shown below. G_2 is a subgraph of G_1 while G_1 and G_3 are isomorphic (use mapping $f = \{t \mapsto a, u \mapsto b, v \mapsto c, w \mapsto d\}$).



G_1 G_2 G_3

The Problem and a Solution

Show that the following problem, called SUBGRAPH ISOMORPHISM, is **NP**-complete: Given two graphs, $G = \langle V_1, E_1\rangle$ and $H = \langle V_2, E_2\rangle$, does G contain a subgraph G' isomorphic to H?

SUBGRAPH ISOMORPHISM is in **NP** because we can first non-deterministically guess the subgraph G' and the isomorphism mapping f in polynomial time and then check (in deterministic polynomial) time that f really is an isomorphism mapping.

We show the **NP**-hardness by reducing from the problem CLIQUE. CLIQUE is the following problem: Given a graph $G = \langle V, E\rangle$ and a (binary coded) integer K, is there a subgraph $\tilde{G} = \langle \tilde{V}, \tilde{E}\rangle$ of G with K or more vertices such that $\tilde{G}$ is complete (for all $\tilde{v}_1, \tilde{v}_2 \in \tilde{V}$, $\langle \tilde{v}_1, \tilde{v}_2\rangle \in \tilde{E}$)? We know that CLIQUE is **NP**-complete (page 190 in the book). Now, given a graph G and an (binary coded) integer K, build a complete graph H such that H has K vertices. Clearly G has a subgraph G' isomorphic to H iff G has a clique with at least K vertices (each clique of size $K' > K$ has a clique of size K as a subgraph). Therefore, the pair $G; K$ is a "yes" instance to CLIQUE iff $G; G'$ is a "yes"-instance to SUBGRAPH ISOMORPHISM. Thus the reduction consists of constructing a complete graph with K vertices, which can clearly be done by using logarithmic auxiliary space.

A Note

Notice that, while SUBGRAPH ISOMORPHISM is **NP**-complete, the problem GRAPH ISOMORPHISM asking whether two graphs are isomorphic is not known to be **NP**-complete nor in **P** (it certainly is in **NP** because we can guess the isomorphism mapping non-deterministically and then verify it in deterministic polynomial time). In fact, GRAPH ISOMORPHISM is one of the main candidates for a language being between languages in **P** and **NP**-complete languages (such languages must exist if $\mathbf{P} \neq \mathbf{NP}$ as will be seen in Chapter 14 in the book).