NP-Complete graph problems - KombSøg 2017

Anders H Pedersen

May 2017

The graph problems and reductions below were introduced by Thomas Dueholm Hansen, at the lecture on NP-Complete graph problems may 3 2017.

1 Graph problems

1.1 INDEPENDENT SET

Given: Graph G = (V,E), target K.
Question: Does there exist $I \subseteq V$ such that $|I| \geq K$ and for all $u, v \in I$ we have $(u, v) \notin E$?

1.2 CLIQUE

Given: Graph G = (V,E), target K.
Question: Does there exist $C \subseteq V$ such that $|C| \geq K$ and for all $u, v \in C$ we have $(u, v) \in E$?

1.3 VERTEX COVER

Given: Graph G = (V,E), budget B.
Question: Does there exist $C \subseteq V$ such that $|C| \leq B$ and for all $(u, v) \in E$ we have $(u \in C \vee v \in C)$?

1.4 MAX CUT

Given: Graph G = (V,E), target K.
Question: Does there exist a cut $(S, V \setminus S)$ of size at least K?

1.5 MAX BISECTION

Given: Graph G = (V,E), target K.
Question: Does there exist a cut $(S, V \setminus S)$ of size at least K, such that $|S| = |V \setminus S|$?

1.6 BISECTION WIDTH

Given: Graph G = (V,E), target K.
Question: Does there exist a cut $(S, V \setminus S)$ of size at most K, such that $|S| = |V \setminus S|$?

1.7 HAMILTONIAN PATH

Given: Graph G = (V,E).
Question: Does G have a path that visits every vertex exactly once?

1.8 TSP

Given: Distance matrix D, target t.
Question: Is there a tour of length at most t that visits every node in the graph defined by D exactly once?

2 Reductions

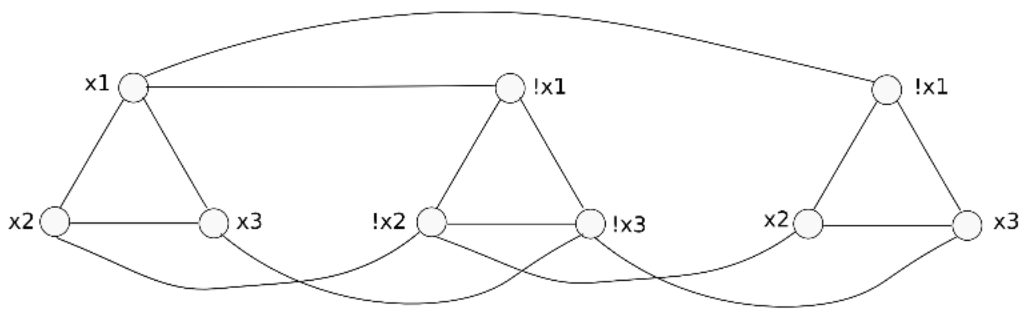
As always, in a reduction $L_1 \leq L_2$, we describe a polynomial time computable function r, such that $\forall x : x \in L_1 \iff r(x) \in L_2$. We then argue that it is indeed polynomial, and that both directions of the bi-implication holds.

2.1 3SAT $\leq$ INDEPENDENT SET

We prove **Theorem 9.4: INDEPENDENT SET is NP-Complete**, by reducing from 3SAT, which we know is NP-Complete.

For this reduction we need a gadget, the triangle. The logic behind this is that, if a graph contains a triangle, then at most one of the nodes can be in the independent set. We restrict the class of graphs we consider, to graphs whose nodes can be partitioned in m disjoint triangles. This ensures that an independent set can contain at most m nodes.

For each of the m clauses in our input CNF formula, we create a triangle where the nodes are labeled are labeled with the literals of the clause. Next we add an edge between two nodes in different triangles if and only if the nodes correspond to opposite literals. For example an edge between x_1 and $\neg x_1$.



Read more in papadimitriou page 188-190.

2.2 INDEPENDENT SET $\leq$ CLIQUE

2.3 INDEPENDENT SET $\leq$ VERTEX COVER

2.4 NAESAT $\leq$ MAX CUT

2.5 3SAT $\leq$ HAMILTONIAN PATH

2.6 HAMILTONIAN PATH $\leq$ TSP