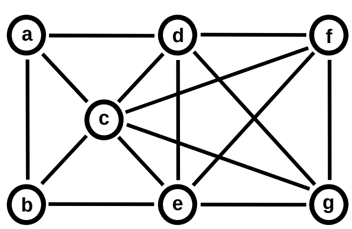
15-453

FORMAL LANGUAGES, AUTOMATA AND COMPUTABILITY

NP-COMPLETENESS II

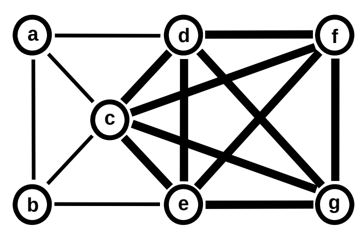
Tuesday April 1

K-CLIQUE



k-clique = complete subgraph of k nodes

K-CLIQUE



k-clique = complete subgraph of k nodes

Assume a reasonable encoding of graphs
(example: the adjacency matrix is reasonable)

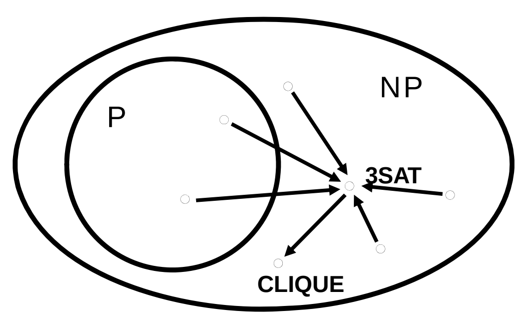
CLIQUE = { (G,k) | G is an undirected graph with a k-clique }

Theorem: CLIQUE is NP-Complete

(1) CLIQUE $\in$ NP

(2) 3SAT $\leq_P$ CLIQUE

CLIQUE is NP-Complete



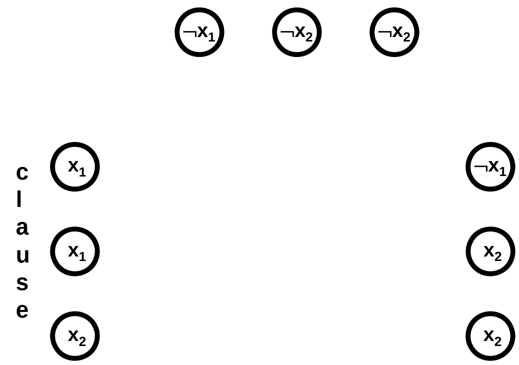
3SAT $\leq_P$ CLIQUE

We transform a 3-cnf formula ϕ into (G,k) such that

$\phi \in$ 3SAT $\Leftrightarrow$ (G,k) $\in$ CLIQUE

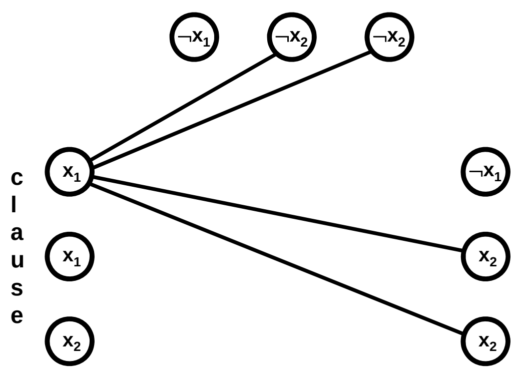
The transformation can be done in time that is polynomial in the length of ϕ

$(x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_2)$



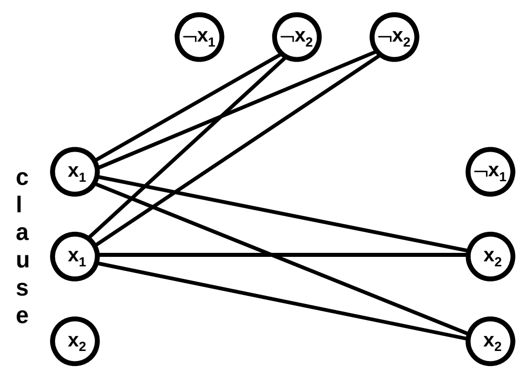
#nodes = 3(# clauses) **k = #clauses**

$(x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_2)$



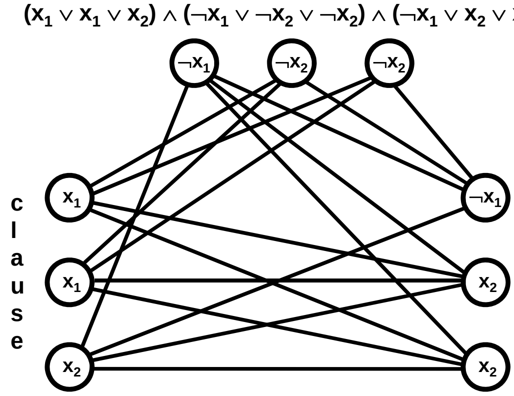
#nodes = 3(# clauses) **k = #clauses**

$(x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_2)$



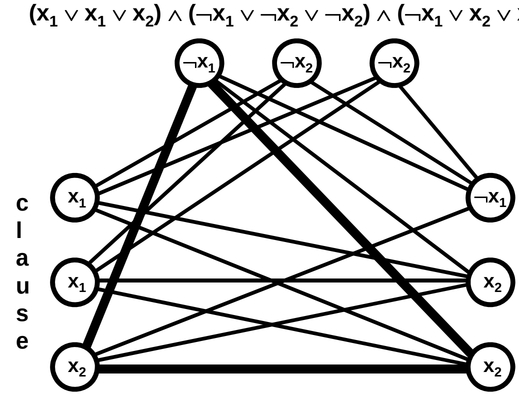
#nodes = 3(# clauses) **k = #clauses**

$(x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_2)$



#nodes = 3(# clauses) **k = #clauses**

$(x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_2)$



#nodes = 3(# clauses) **k = #clauses**

$3SAT \leq_P$ CLIQUE

We transform a 3-cnf formula ϕ into (G,k) such that

$\phi \in 3SAT \Leftrightarrow (G,k) \in$ CLIQUE

If ϕ has m clauses, we create a graph with m clusters of 3 nodes each, and set k=m
Each cluster corresponds to a clause.
Each node in a cluster is labeled with a literal from the clause.

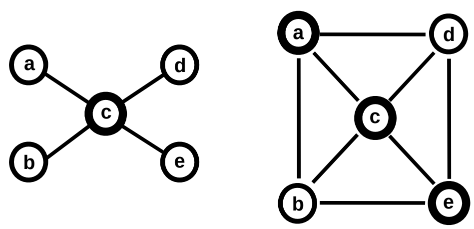
We do not connect any nodes in the same cluster

We connect nodes in different clusters whenever they are not contradictory

The transformation can be done in time that is polynomial in the length of ϕ

$(x_1 \lor x_1 \lor x_1) \land (\neg x_1 \lor \neg x_1 \lor x_2) \land$
$(x_2 \lor x_2 \lor x_2) \land (\neg x_2 \lor \neg x_2 \lor x_1)$

VERTEX COVER



vertex cover = set of nodes that cover all edges

VERTEX-COVER = { (G,k) | G is an undirected graph with a k-node vertex cover }

Theorem: VERTEX-COVER is NP-Complete

(1) VERTEX-COVER $\in$ NP

(2) $3SAT \leq_P$ VERTEX-COVER

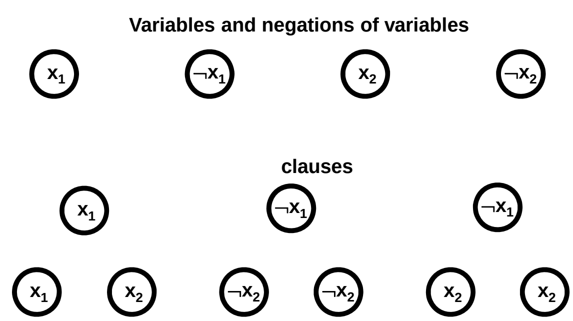
$3SAT \leq_P$ VERTEX-COVER

We transform a 3-cnf formula ϕ into (G,k) such that

$\phi \in 3SAT \Leftrightarrow (G,k) \in$ VERTEX-COVER

The transformation can be done in time polynomial in the length of ϕ

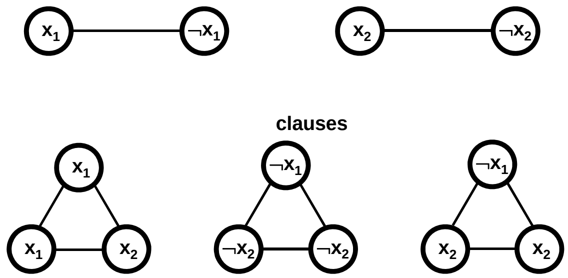
$(x_1 \lor x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_2)$



ϕ Satisfiable => put "true" literals on top in vertex cover
For each clause, pick a true literal and put other 2 in vertex cover

$(x_1 \lor x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_2)$

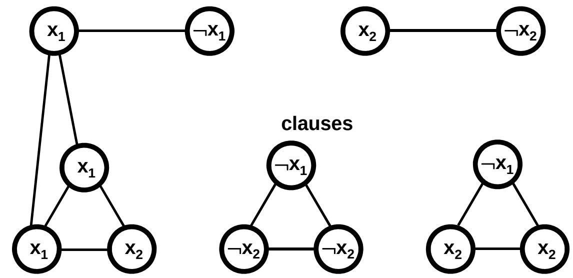
Variables and negations of variables



ϕ Satisfiable => put "true" literals on top in vertex cover
For each clause, pick a true literal and put other 2 in vertex cover

$(x_1 \lor x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_2)$

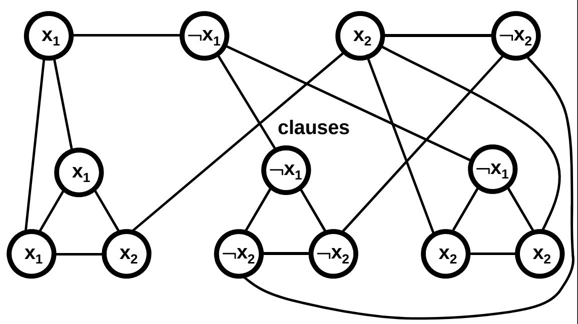
Variables and negations of variables



ϕ Satisfiable => put "true" literals on top in vertex cover
For each clause, pick a true literal and put other 2 in vertex cover

$(x_1 \lor x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_2)$

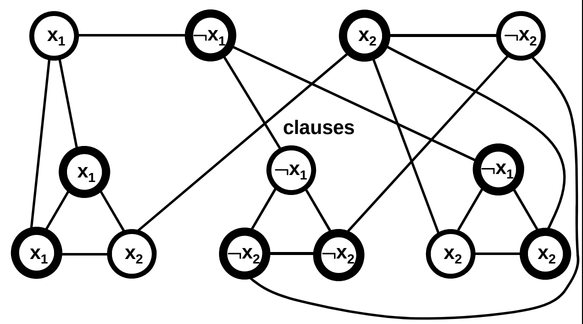
Variables and negations of variables



k = 2(#clauses) + (#variables)

$(x_1 \lor x_1 \lor x_2) \land (\neg x_1 \lor \neg x_2 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_2)$

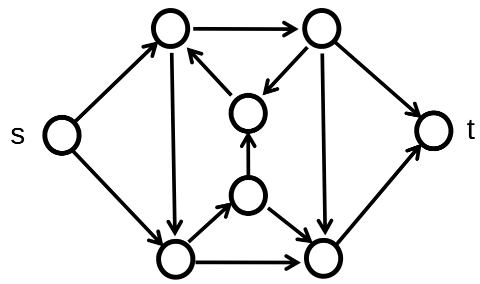
Variables and negations of variables



k = 2(#clauses) + (#variables)

$(x_1 \lor x_1 \lor x_1) \land (\neg x_1 \lor \neg x_1 \lor x_2) \land$
$(x_2 \lor x_2 \lor x_2) \land (\neg x_2 \lor \neg x_2 \lor x_1)$

HAMILTON PATH



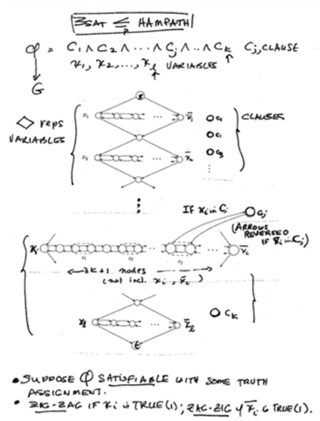
HAMPATH = { (G,s,t) | G is an directed graph with a Hamilton path from s to t}

Theorem: HAMPATH is NP-Complete

(1) HAMPATH ∈ NP

(2) 3SAT $\leq_P$ HAMPATH

Proof is in Sipser, Chapter 7.5



$(x_1 \vee x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_2)$

WWW.FLAC.WS