Polynomial-Time Reduction

Lecture 38
Section 14.6

Robb T. Koether

Hampden-Sydney College

Fri, Dec 2, 2016

Outline

1. Polynomial-Time Reduction
2. The Decision Problem 3SAT
3. Reduction of 3SAT to CLIQ
4. Reduction of CLIQ to VC
5. Some Theorems
6. Assignment

Polynomial-Time Reduction

Definition (Polynomial-Time Reduction)

A language L_1 is reducible in polynomial time to a language L_2 if there is a deterministic Turing machine M that computes a function $f: \Sigma^* \to \Sigma^*$ with the properties that

- For all $w \in \Sigma^*$, $w \in L_1$ if and only if $f(w) \in L_2$.
- For all $w \in L_1$, M computes $f(w)$ in polynomial time.

Language of a Problem

Definition (Language of a Problem)

Given a decision problem A and an encoding of instances of A, the language of A is the set of encodings of all instances of A for which the answer is "yes."

- Thus, we can speak of a Turing machine *reducing* one decision problem to another decision problem in polynomial time.

Language of a Problem

Definition (Language of a Problem)

Given a decision problem A and an encoding of instances of A, the language of A is the set of encodings of all instances of A for which the answer is "yes."

- Thus, we can speak of a Turing machine *reducing* one decision problem to another decision problem in polynomial time.
- For example,

$$L(\text{COMP}) = \{100, 110, 1000, 1001, 1010, 1100, \ldots\}.$$

Outline

Example (The Decision Problem 3SAT)

- The decision problem 3SAT is like the problem SAT except that each clause must contain exactly 3 literals (3CNF).
- Any instance of SAT may easily be converted into an instance of 3SAT in polynomial time.

Reducing SAT to 3SAT

Example (Reducing SAT to 3SAT)

- If a clause has only 1 or 2 literals, we add new literals as follows:

Reducing SAT to 3SAT

Example (Reducing SAT to 3SAT)

- If a clause has only 1 or 2 literals, we add new literals as follows:

$$x_1 \vee x_2 = (x_1 \vee x_2 \vee y) \wedge (x_1 \vee x_2 \vee \overline{y}),$$

Reducing SAT to 3SAT

Example (Reducing SAT to 3SAT)

- If a clause has only 1 or 2 literals, we add new literals as follows:

$$x_1 \vee x_2 = (x_1 \vee x_2 \vee y) \wedge (x_1 \vee x_2 \vee \overline{y}),$$
$$x_1 = (x_1 \vee y \vee z) \wedge (x_1 \vee y \vee \overline{z}) \wedge (x_1 \vee \overline{y} \vee z) \wedge (x_1 \vee \overline{y} \vee \overline{z}).$$

Reducing SAT to 3SAT

Example (Reducing SAT to 3SAT)

- If a clause has only 1 or 2 literals, we add new literals as follows:

$$
\begin{aligned}
x_1 \vee x_2 &= (x_1 \vee x_2 \vee y) \wedge (x_1 \vee x_2 \vee \overline{y}),\\
x_1 &= (x_1 \vee y \vee z) \wedge (x_1 \vee y \vee \overline{z}) \wedge (x_1 \vee \overline{y} \vee z) \wedge (x_1 \vee \overline{y} \vee \overline{z}).
\end{aligned}
$$

- If a clause has more than 3 literals, we do something similar.

Reducing SAT to 3SAT

Example (Reducing SAT to 3SAT)

- If a clause has only 1 or 2 literals, we add new literals as follows:

$$\begin{aligned} x_1 \vee x_2 &= (x_1 \vee x_2 \vee y) \wedge (x_1 \vee x_2 \vee \overline{y}), \\ x_1 &= (x_1 \vee y \vee z) \wedge (x_1 \vee y \vee \overline{z}) \wedge (x_1 \vee \overline{y} \vee z) \wedge (x_1 \vee \overline{y} \vee \overline{z}). \end{aligned}$$

- If a clause has more than 3 literals, we do something similar.
- For example,

$$x_1 \vee x_2 \vee x_3 \vee x_4 = (x_1 \vee x_2 \vee y) \wedge (x_3 \vee x_4 \vee \overline{y}).$$

Outline

Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)

- Let f be a Boolean expression in 3CNF.
- Create a graph G by the following two steps.

(1) For each clause, create a group of nodes labeled with the literals in that clause.

Example (Reducting 3SAT to CLIQ)

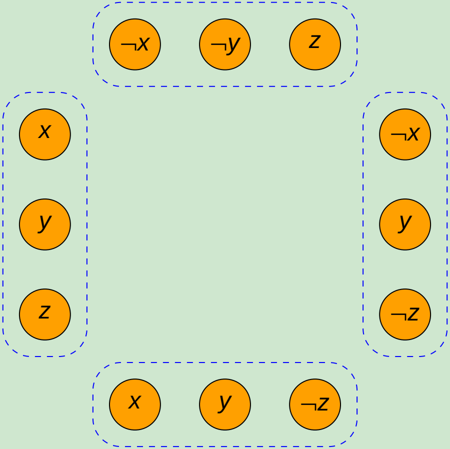
- For example, let

$$e = (x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z) \wedge (x \vee y \vee \neg z).$$

- Then there are four groups of three nodes each.

Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



Reduction of 3SAT to CLIQ

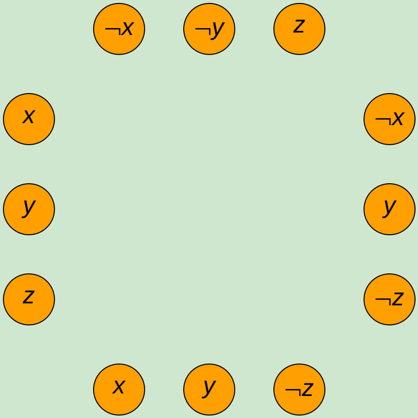
Example (Reducting 3SAT to CLIQ)

(2) Connect each node in one group with every node in the other groups with which it is logically compatible.

- That is, for every variable x, connect x with everything except $\neg x$ in every other clause.
- Do this for each group.

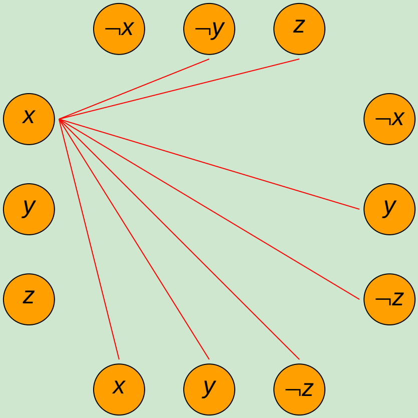
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



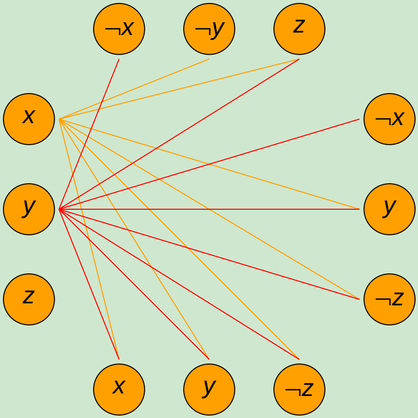
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



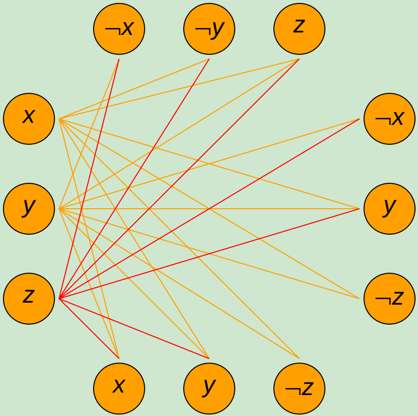
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



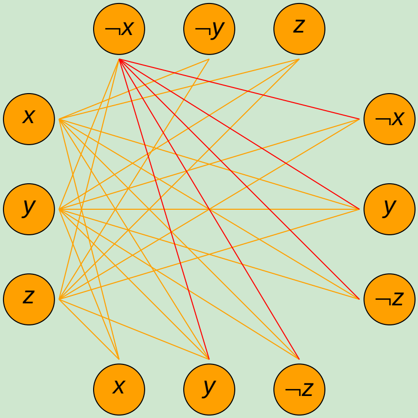
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



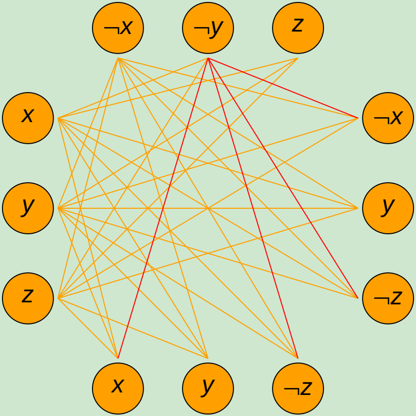
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



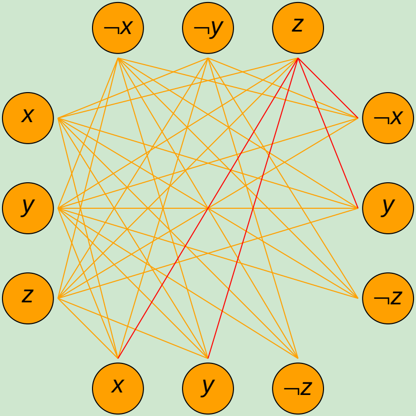
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



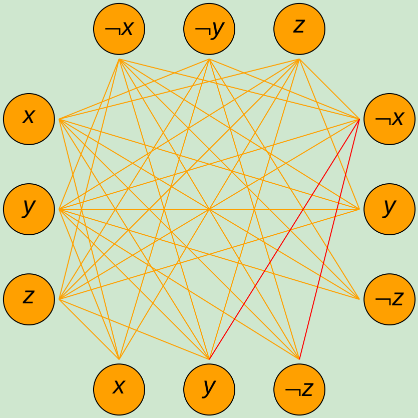
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



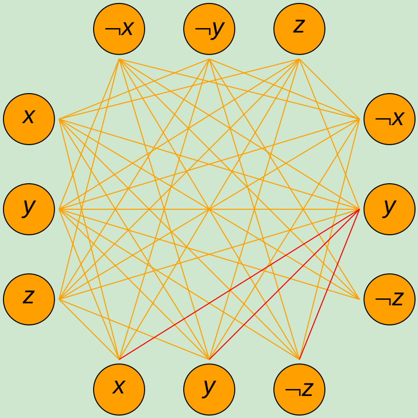
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



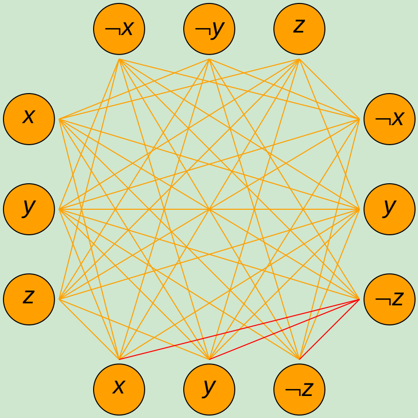
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



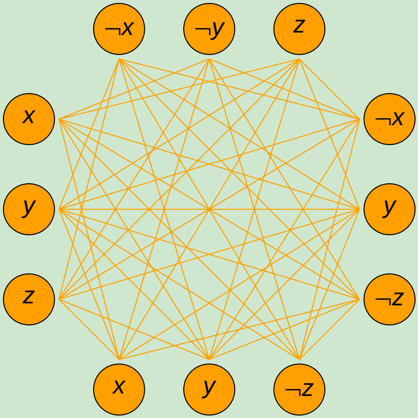
Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)



Reduction of 3SAT to CLIQ

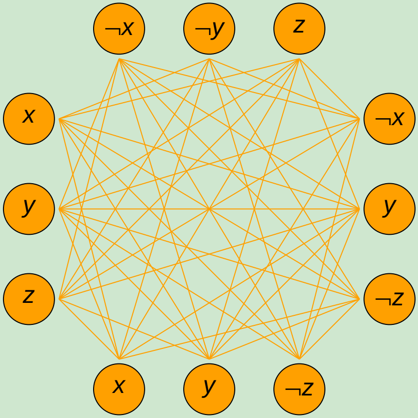
Example (Reducting 3SAT to CLIQ)

- Let k be the number of clauses in the expression. ($k = 4$)
- We now ask, does the graph have a clique of size k?

Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)

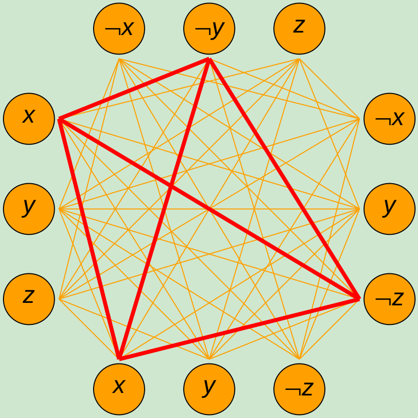
- Does this graph have a clique of size 4?



Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)

- Yes it does, namely $\{x, \neg y, \neg z\}$.



Reduction of 3SAT to CLIQ

Example (Reducting 3SAT to CLIQ)

- This clique gives us values for x, y, and z that will satisfy the expression.
- Namely, x is true, y is false, and z is false.
- This shows that "yes" to CLIQUE implies "yes" to 3SAT.
- It is also easy to see that "no" to CLIQUE implies "no" to 3SAT.
- It is also the case that this reduction can be done in polynomial time.

Outline

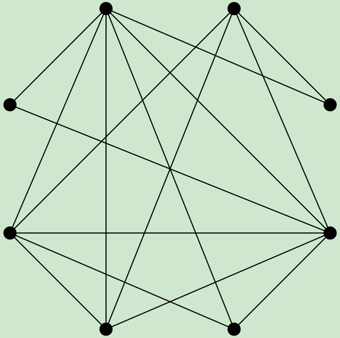
Reduction of CLIQ to VC

Example (Reduction of CLIQ to VC)

- Given a graph G and an integer k, we reduce the Vertex Cover Problem (VC) to CLIQ.
  - Let $\overline{G}$ be the complementary graph.
  - That is, e is an edge of $\overline{G}$ if and only e is *not* an edge of G.
  - Let n be the number of vertices in G.
  - Then solve CLIQUE for $\overline{G}$ and the integer $n - k$.

Reduction of CLIQ to VC

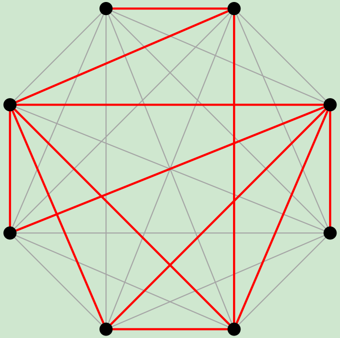
Example (Reduction of CLIQ to VC)



Find a vertex cover of G of size $k = 4$

Reduction of CLIQ to VC

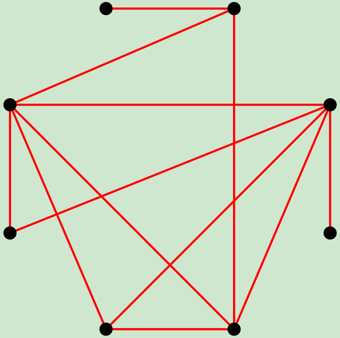
Example (Reduction of CLIQ to VC)



Consider the complementary graph $\overline{G}$ ($O(n^2)$)

Reduction of CLIQ to VC

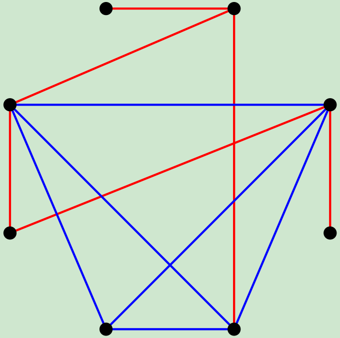
Example (Reduction of CLIQ to VC)



Consider the complementary graph $\overline{G}$

Reduction of CLIQ to VC

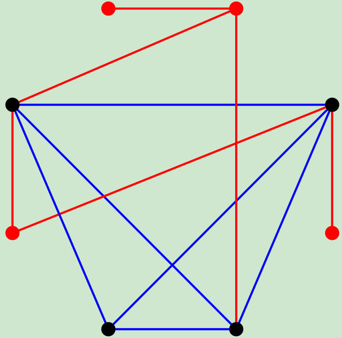
Example (Reduction of CLIQ to VC)



Find a clique of $\overline{G}$ of size $n - k = 4$ $(O(n^i))$

Reduction of CLIQ to VC

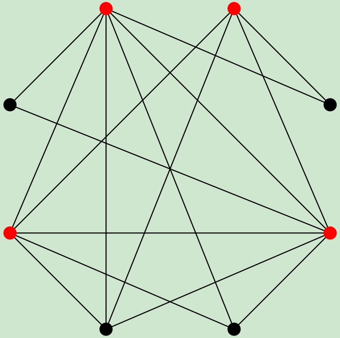
Example (Reduction of CLIQ to VC)



The complementary vertices the clique ($O(n)$)...

Reduction of CLIQ to VC

Example (Reduction of CLIQ to VC)



…form a vertex cover of G of size $k = 4$

Outline

Some Theorems

Theorem

If a problem A is reducible in polynomial time to SAT, then $A \in \mathbf{NP}$.

Some Theorems

Theorem

If a problem A is reducible in polynomial time to SAT, then $A \in \mathbf{NP}$.

Theorem

If a problem A can be reduced to SAT in polynomial time, then it can be reduced to 3SAT in polynomial time.

Some Theorems

Theorem

If a problem A is reducible in polynomial time to SAT, then $A \in$ **NP**.

Theorem

If a problem A can be reduced to SAT in polynomial time, then it can be reduced to 3SAT in polynomial time.

Theorem

If a problem A can be reduced to 3SAT in polynomial time, then it can be reduced to CLIQ in polynomial time.

Some Theorems

Theorem

If a problem A is reducible in polynomial time to SAT, then $A \in \mathbf{NP}$.

Theorem

If a problem A can be reduced to SAT in polynomial time, then it can be reduced to 3SAT in polynomial time.

Theorem

If a problem A can be reduced to 3SAT in polynomial time, then it can be reduced to CLIQ in polynomial time.

Theorem

If a problem A can be reduced to CLIQ in polynomial time, then it can be reduced to VC in polynomial time.

Some Theorems

Theorem

The Vertex Cover Problem is in **NP**.

Proof.

- Let G be a graph with n vertices and let k be an integer.
- Generate a solution.
  - Nondeterministically, select a set C of vertices of size k ($O(n)$).
- Verify the solution.
  - For each edge e, check whether e is incident to a vertex in C ($O(n)$).
  - There are at most $\frac{1}{2}n(n-1) \in O(n^2)$ edges in G, so this can be done in $O(n^3)$ time.
- Therefore, VC $\in$ **NP**.

□

Outline

Assignment

Homework

- Section 14.5 Exercises 3, 4, 5, 8.
- Section 14.6 Exercises 1, 4, 5.