Exercise: Let $G = (V, E)$ be an undirected graph

11/12/2009
E 8.1

A vertex cover C of G is a subset of the vertices V s.t. every edge of G touches at least one of the nodes of C.

The vertex cover problem:

input: - graph $G = (V, E)$
- integer k

output: yes iff G has a vertex cover of size $\leq k$

Vertex-cover is NP-complete:

Proof:

in NP: easy

- guess a subset C of V of size $\leq k$
- check in poly-time that it is a vertex-cover

NP-hard by reduction from 3-SAT

We define a poly-time reduction R that:

- takes as input a 3-CNF formula F
- constructs a graph $G = (V, E)$ and an integer k

such that:

F is satisfiable $\Leftrightarrow$ G admits a vertex cover with k nodes

Let $F = C_1 \cdot \dots \cdot C_m$ be a 3-CNF formula over variables $\{x_1, \dots, x_n\}$

We construct $G = (V, E)$ as constituted by various components

E 8.2

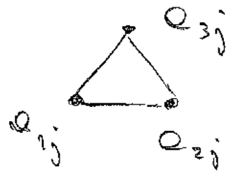
- For each variable x_i, we have a truth-setting component $T_i = (V_i, E_i)$ with $V_i = \{x_i, \bar{x}_i\}$

$$E_i = \{\{x_i, \bar{x}_i\}\}$$

undirected edge

note: at least one of x_i, $\bar{x}_i$ will be in every vertex cover to cover $\{x_i, \bar{x}_i\}$

- For each clause C_j in F we have a satisfaction testing component $S_j = (V_j', E_j')$



note: at least two of V_j' will be in every vertex cover to cover E_j'

- We have a communication component, which is the only part that depends on which literals are in which clauses

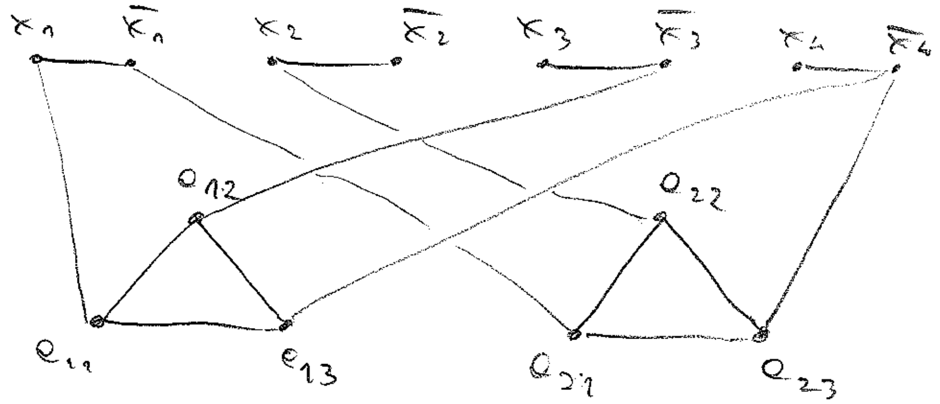
Let $C_j = l_{1j} + l_{2j} + l_{3j}$

Then we have $E_j'' = \{\{a_{1j}, l_{1j}\}, \{a_{2j}, l_{2j}\}, \{a_{3j}, l_{3j}\}\}$

We then set $K = n + 2m$

\# variables   \# clauses

Example: $F = (x_1 + \overline{x}_3 + \overline{x}_4) \cdot (\overline{x}_1 + x_2 + \overline{x}_4)$ E 8.3



$k = n + 2m = 4 + 2 \cdot 2 = 8$

We show that F is satisfiable $\Longleftrightarrow$ G has a vertex cover of size $\leq k$

"$\Longleftarrow$" Let $V' \subseteq V$ be a vertex cover for G with $|V'| \leq k$.

We said that V' contains at least
- one vertex for each variable
- 2 vertices for each clause

This is already $k = n + 2m$

$\Longrightarrow$ at least is actually exactly

We use V' to obtain the truth assignment γ

we set $x_i = \text{true}$ if $x_i \in V'$

$x_i = \text{false}$ if $x_i \notin V'$

To show that γ is a truth assignment that satisfies F, we exploit that all arcs of the communication components are covered by V'.

Consider a clause $C_j = l_{1j} + l_{2j} + l_{3j}$

- two of the arcs in E''_j are covered by the choice of two among e_{1j}, e_{2j}, e_{3j} in V'.

W.l.o.g., let these be e_{1j}, e_{2j}

- the third arc is then covered by the literal l_{3j} (connected to e_{3j}) which has to be in V'

Since, by definition of γ, $l_{3j} = \text{true}$, C_j is satisfied

"$\Rightarrow$" Let γ be a truth assignment that satisfies F.

We define a subset $V' \subseteq V$ as follows

- $x_i \in V'$ iff $\gamma(x_i) = \text{true}$
  $\overline{x}_i \in V'$ iff $\gamma(x_i) = \text{false}$

Since γ satisfies F, for each communication component $E''_j = \{\{e_{1j}, l_{1j}\}, \{e_{2j}, l_{2j}\}, \{e_{3j}, l_{3j}\}\}$, one of the three edges $\{e_{ij}, l_{ij}\}$ is covered in V' by l_{ij}. W.l.o.g., let $i = 1$. Then $\{e_{2j}, l_{2j}\}$, $\{e_{3j}, l_{3j}\}$ can be covered by having $e_{2j} \in V'$ and $e_{3j} \in V'$.

We get that V' contains $n + 2m$ vertices.

Exercise (10.3.2): The problem 4TA - SAT is defined as follows:

Give a prop. formula E, does E have at least 4 satisfying truth assignments?

Show that 4TA - SAT is NP-complete:

Proof:

1) 4 TA - SAT is in NP

We devise a non-deterministic poly-time algorithm.

1) Guess 4 truth-assignments T_1, T_2, T_3, T_4

2) Check that T_1, T_2, T_3, T_4 all satisfy E.

Note that both steps require time polynomial in the size of E

2) 4TA - SAT is NP-hard

We show this by reducing SAT to 4TA - SAT.

Let E be a prop. formula, and let $x_1, \dots, x_n$ be all variables in E.

We construct a new formula E' s.t.

$$E \in SAT \iff E' \in 4TA\text{-}SAT$$

Let y_1, y_2 be two new variables. Then

$$E' = E \vee ((x_1 \wedge x_2 \wedge \dots \wedge x_n) \wedge ((y_1 \wedge y_2) \vee (y_1 \wedge \overline{y_2}) \vee (\overline{y_1} \wedge y_2)))$$

Consider the truth assignments for $x_1, \dots, x_n, y_1, y_2$

	x_1	$\cdot x_j$	x_n	y_1	y_2	Case 1 $E \notin SAT$: E	Case 1 $E \notin SAT$: E'	Case 2 $E \in SAT$: E	Case 2 $E \in SAT$: E'
1)	T	...	T	T	T	F	T	?	T
2)	T	...	T	T	F	F	T	?	T
3)	T	...	T	F	T	F	T	?	T
4)	T	...	T	F	F	F	F	⋮	⋮
⋮	...	F	...		...	F	F	E	T
2^{n+2}		⋮				F	F	⋮	⋮
							$=3$		≥ 4

Alternative solution:

$$E' = E \wedge (y_1 \vee y_2 \vee y_3)$$

- If E is unsatisfiable, then E' is unsatisfiable, and hence $E' \notin$ 47A-SAT
- If E is satisfiable, then E' has at least 7 satisfying truth assignments: these are obtained by combining
  - a TA for $x_1, \dots, x_n$ satisfying E with
  - the 7 TAs for y_1, y_2, y_3 satisfying $y_1 \vee y_2 \vee y_3$

Exercise 11.1.1 b)

Consider the problem FALSE-SAT:

Given a boolean expression E that is false when all its variables are made false, is there some other truth assignment that makes E false, besides all-false?

Decide whether the problem is in NP or coNP.

Describe its complement.

If the problem or its complement is NP-complete, prove it.

Proof:

The problem is NP-complete.

- In NP: given a boolean expression E, we need to check:
  1) that E is false when all variables are assigned false
  2) that there is some other truth-assignment making E false

(1) can be done in poly-time by a DTM

(2) can be done in poly-time by a NTM

guess a truth-assignment T different from all false, and answer yes if under T, E evaluates to false

- NP-hard: by a reduction from SAT

Let E be a boolean expression with variables $x_1, \ldots, x_n$.

We construct an expression E' s.t. $E \in SAT$ iff $E' \in$ FALSE-SAT

1) Test if E is true when all variables are false (polynomial)

If so, $E \in SAT$, and we convert it to a fixed expression that is in FALSE-SAT, e.g. $x \wedge y$.

2) Otherwise, let E' be $\neg E \wedge (x_1 \vee x_2 \vee \cdots \vee x_n)$. E 8.8

Clearly, the reduction is poly-time.

We have that E' is false when all of $x_1, \cdots, x_n$ are false.

Notice that in case (2), E is false when all variables are false.

Hence, if $E \in SAT$, then it is satisfied by a truth assignment T different from all-false.

Thus, $\neg E$ is made false by T, and $E' \in$ FALSE-SAT!

Conversely, if $E' \in$ FALSE-SAT, then since $x_1 \vee \cdots \vee x_n$ is false only for the all-false truth assignment, there must be some other truth-assignment T that makes $\neg E$ false. Then T makes E true, and $E \in SAT$.