

Exercise: Reduction from 3-SAT to CSAT

1/12/2010
E 8.1

(see textbook 10.3.4)

Given a CNF formula $E = C_1 \cdot C_2 \cdots C_k$

with each $C_i = \sum_{j=1}^{k_i} l_{ij}$,

we construct a 3-CNF formula F as follows.

For each clause C_i of E

1) if $C_i = (l)$ (i.e., a single literal)

introduce two new variables u, v , and replace C_i by 4 clauses

$$(l + u + v).$$

$$(l + u + \bar{v}).$$

$$(l + \bar{u} + v).$$

$$(l + \bar{u} + \bar{v}).$$

Since u, v appear in all 4 combinations, the 4 clauses can be satisfied only if l is true

2) if $C_i = (l_1 + l_2)$

introduce a new variable z , and replace C_i by 2 clauses

$$(l_1 + l_2 + z).$$

$$(l_1 + l_2 + \bar{z}).$$

as in 1

3) if $C_i = (l_1 + l_2 + l_3)$, just leave it

4) if $C_i = (l_1 + l_2 + \cdots + l_m)$ with $m \geq 4$

introduce y_1, y_2, \dots, y_{m-3} and replace C_i by

$$(l_1 + l_2 + y_1) \cdot (l_3 + \bar{y}_1 + y_2) \cdot (l_4 + \bar{y}_2 + y_3) \cdots \\ + (l_{m-2} + \bar{y}_{m-4} + y_{m-3}) \cdot (l_{m-1} + l_m + \bar{y}_{m-3})$$

- An assignment T satisfying E makes at least one literal of C_i true. Let it be l_j .

Then, by making y_{j-1}, \dots, y_{j-2} true and y_{j-1}, \dots, y_{j-3} false,

we satisfy all clauses replacing C_i .

Thus we can extend T to satisfy F .

- Conversely, if T makes all l_j of C_i false, then not all new clauses can be satisfied.

Why? each y_j can make at most 1 clause true, but there are $m-2$ clauses and $m-3$ y_j 's.

The 3CNF formula F is linear in E and can be constructed in linear time

We get: $\text{CSAT} \leq_{\text{poly}} \text{3-SAT}$

\Rightarrow from CSAT NP-hard, we get 3-SAT NP-hard

We also know $\text{3-SAT} \in \text{P}$ (since $\text{SAT} \in \text{P}$)

\Rightarrow 3-SAT is NP-complete

Exercise: 2SAT is in P

Idea: we show that 2SAT can be encoded as a graph reachability problem, and then use an algorithm for graph reachability

1) Encoding of 2SAT as a directed graph reachability problem

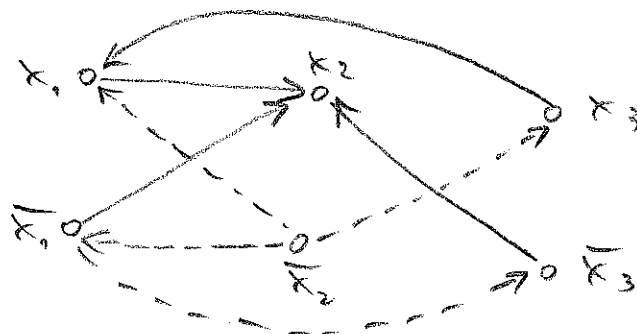
Let Φ be an instance of 2SAT. We define a graph $G(\Phi)$ as follows:

- one node for each variable and for each negated variable

- for each clause $\alpha \vee \beta$ two edges $\bar{\alpha} \rightarrow \beta$
 $\bar{\beta} \rightarrow \alpha$

(note: $\alpha \vee \beta \equiv \bar{\alpha} \rightarrow \beta \equiv \bar{\beta} \vee \alpha$)

Example: $(x_1 \vee x_2)$.
 $(x_1 \vee \bar{x}_3)$.
 $(\bar{x}_1 \vee x_2)$.
 $(x_2 \vee x_3)$



Then Φ is unsatisfiable iff there is a variable x such that $G(\Phi)$ contains two paths $x \rightarrow \dots \rightarrow \bar{x}$

" \Leftarrow " Suppose that Φ has a satisfying truth assignment T . Assume that $T(x) = \text{true}$ (a similar argument holds for $T(x) = \text{false}$)

Since $T(x) = \text{true}$ and $T(\bar{x}) = \text{false}$, and there is a path $x \rightarrow \dots \rightarrow \bar{x}$, there must be an edge $\alpha \rightarrow \beta$ along this path with $T(\alpha) = \text{true}$ and $T(\beta) = \text{false}$.

However, since $\alpha \rightarrow \beta$ is an edge of $G(\Phi)$, it follows that $\neg\beta$ is a clause of Φ . This clause is not satisfied by T , which is a contradiction.

" \Rightarrow " Let $G(\Phi)$ be a graph that does not contain any node x with $x \xrightarrow{\dots} \bar{x}$

We construct from such a graph a satisfying truth assignment T

Repeat the following step as often as possible:

Choose a node α such that

- $T(\alpha)$ is not yet defined, and
- there is no path $\alpha \xrightarrow{*} \bar{\alpha}$

For every node β that is reachable from α

1) set $T(\beta) = \text{true}$

2) set $T(\bar{\beta}) = \text{false}$

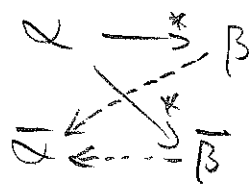
(Note: 2) means to assign false to all predecessors of $\bar{\alpha}$)



Observe: 1) the truth assignment T is well defined, i.e., we never have both $T(\beta) = \text{true}$ and $T(\bar{\beta}) = \text{true}$ or $T(\beta) = \text{false}$ and $T(\bar{\beta}) = \text{false}$

T would not be well defined, if we had both $\alpha \xrightarrow{*} \beta$ and $\alpha \xrightarrow{*} \bar{\beta}$ (for some β)

But this cannot happen, since we would have



hence, we would have $\alpha \xrightarrow{*} \bar{\alpha}$

2) We assign to all nodes a truth value, since there is no path $\alpha \xrightarrow{*} \bar{\alpha}$

3) The truth assignment satisfies all clauses of F , since each clause corresponds to an implication, and there is no $\alpha \xrightarrow{*} \bar{\alpha}$.

Exercise: Let $G = (V, E)$ be an undirected graph E 8.5

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 $\{x_1, \dots, x_n\}$

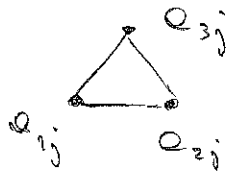
We construct $G = (V, E)$ as constituted by various components

- 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 $C_j \in \mathcal{F}$ 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

$$\text{let } C_j = l_{1j} + l_{2j} + l_{3j}$$

$$\text{then we have } E''_j = \{ \{e_{1j}, l_{1j}\}, \{e_{2j}, l_{2j}\}, \{e_{3j}, l_{3j}\} \}$$

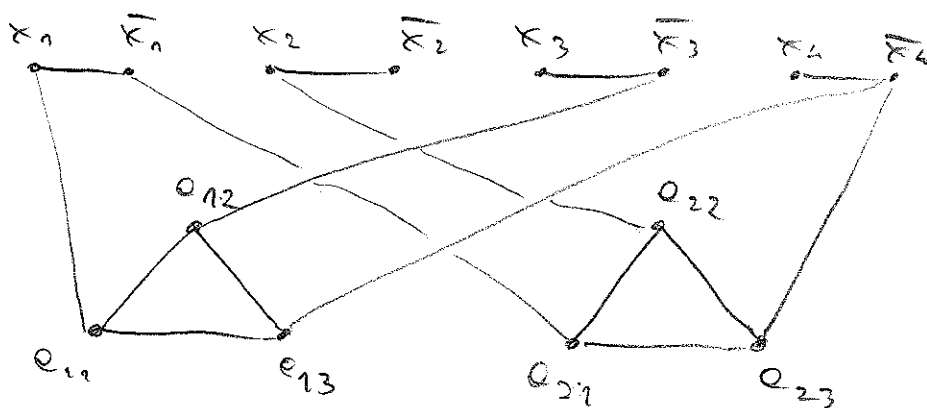
We then set $K = n + 2m$

↑ # variables

↑ # clauses

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

E 8.7



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

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

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

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

This is already $k = n + 2m$

\Rightarrow 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 $\bar{x}_i \in V'$

To show that γ is a truth assignment that satisfies F , we explain that all clauses 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 true among e_{1j}, e_{2j}, e_{3j} in V' .

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

- the third arc is then covered by the literal l_{3j} E 8.8
(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 $\gamma(x_i) = \text{true}$

$\bar{x}_i \in V'$ 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.