Algorithms and Complexity

Spring 2018
Aaram Yun

This page is intentionally left blank

Today

- Concrete NP-complete problems

NP-completeness of 3SAT

- $\mathtt{3SAT} = \{\phi : \phi \text{ is a satisfiable 3CNF}\}$
- $R_{\mathtt{3SAT}} = \{(\phi, \tau) : \phi(\tau) = 1 \text{ and } \phi \text{ is a 3CNF}\}$
- Theorem $\mathtt{3SAT}$ is $\mathcal{NP}$-complete and $R_{\mathtt{3SAT}}$ is $\mathcal{PC}$-complete
  - Proof: reduce from $\mathtt{SAT}$

Set cover $(\{S_1, \ldots, S_m\}, K)$

- **SC** problem: given a collection of finite sets $S_1, \ldots, S_m$ and an integer K, decide whether there exist (at most) K sets that cover $\cup_{i=1}^{m} S_i$
  - In other words, indices $i_1, \ldots, i_K$ such that $\cup_{j=1}^{K} S_{i_j} = \cup_{i=1}^{m} S_i$

$$X = \bigcup_{j=1}^{m} S_j$$

$(S_1, \ldots, S_m, K)$

$(\{S_{i,t}, S_{i,f}\}_{i=1}^{n}, n)$; ???

not good enough

Set cover

Design a reduction from SAT to SC

$S_{i,t}, S_{i,f} \subseteq \{1, 2, \cdots, m\}$

- **SC** is NP-complete
  - Need to make an instance of **SC**, when given a CNF $\phi(x_1, \ldots, x_n) = C_1 \wedge \cdots \wedge C_m$

$C_j = \cdots \vee x_i \vee \cdots$

- $S_{i,\mathtt{t}} = \{j : x_i = 1 \text{ makes } C_j = 1\}$

$C_j = \cdots \vee \overline{x_i} \vee \cdots$

- $S_{i,\mathtt{f}} = \{j : x_i = 0 \text{ makes } C_j = 1\}$

$S_{i,t} \cup S_{i,f} = \{j \mid x_i \text{ occurs inside } C_j\}$

$\rightarrow \bigcup_{i=1}^{n} S_{i,t} \cup S_{i,f} = \{1, \cdots, m\}$

$C_k = \cdots \vee x_i \vee \cdots$

or

$C_k = \cdots \vee \overline{x_i} \vee \cdots$

$\rightarrow k \in S_{i,t} \cup S_{i,f}$

We cannot directly use $(\{S_{i,t}, S_{i,f}\}_{i=1}^{n}, n)$.

Define $S_{i,t}' := S_{i,t} \cup \{i'\}$ $(\{S_{1,t}, S_{1,f}, S_{2,t}, S_{2,f}, \dots, S_{n,t}, S_{n,f}\}, n)$

$S_{i,f}' := S_{i,f} \cup \{i'\}$

$$\bigcup_{i=1}^{n} S_{i,t}' \cup S_{i,f}' = \{1, 2, \dots, m\} \cup \{1', 2', \dots, n'\}$$

Now consider $(\{S_{i,t}', S_{i,f}'\}_{i=1}^{n}, n)$

Since only $S_{i,t}', S_{i,f}'$ has i', a possible solution should have at least one of $S_{i,t}'$ or $S_{i,f}'$, for each i.

If $S_{i,t}'$ is chosen, then we set $x_i = 1$.

If $S_{i,f}'$ is chosen, then we set $x_i = 0$.

Vertex cover

- `VC`: given a simple graph $G = (V, E)$ and an integer K, decide whether there exists a set of (at most) K vertices that is incident to all graph edges.

Vertex cover

- **`VC`** is NP-complete

Exercise

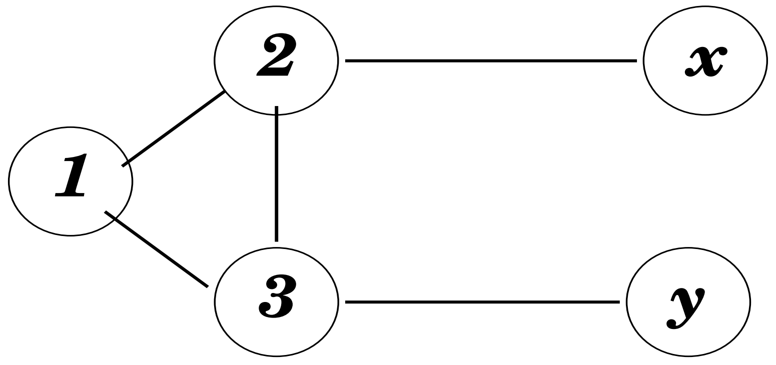
Graph colorability

- **G3C**: given a graph G, decide whether it is 3-colorable, that is, whether it is possible to color each vertex in one of three colors, so that no adjacent vertices are colored with the same color
- **G3C** is NP-complete

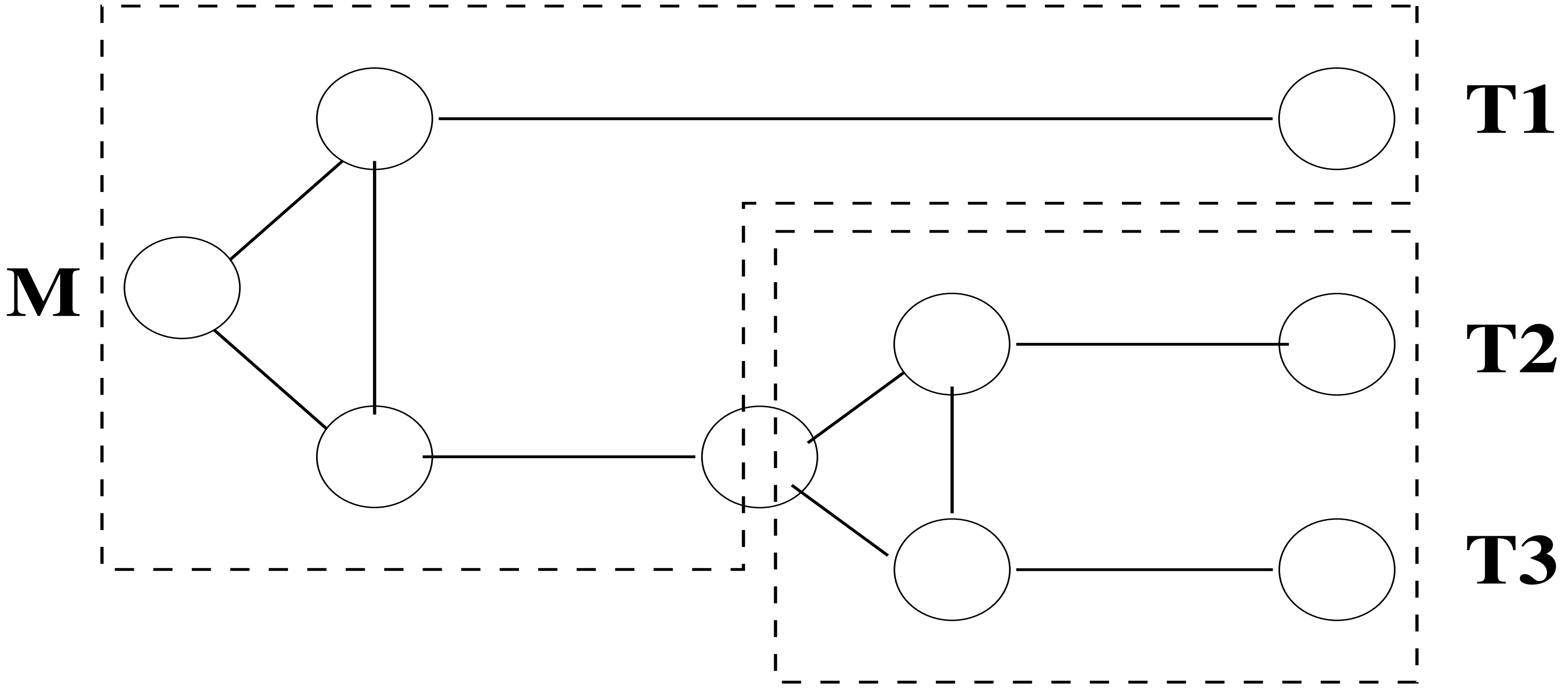
3SAT to G3C.

$\phi \rightsquigarrow$ graph G.

t, f, g : three colors



When $x=y$, necessarily $x=y=1$.

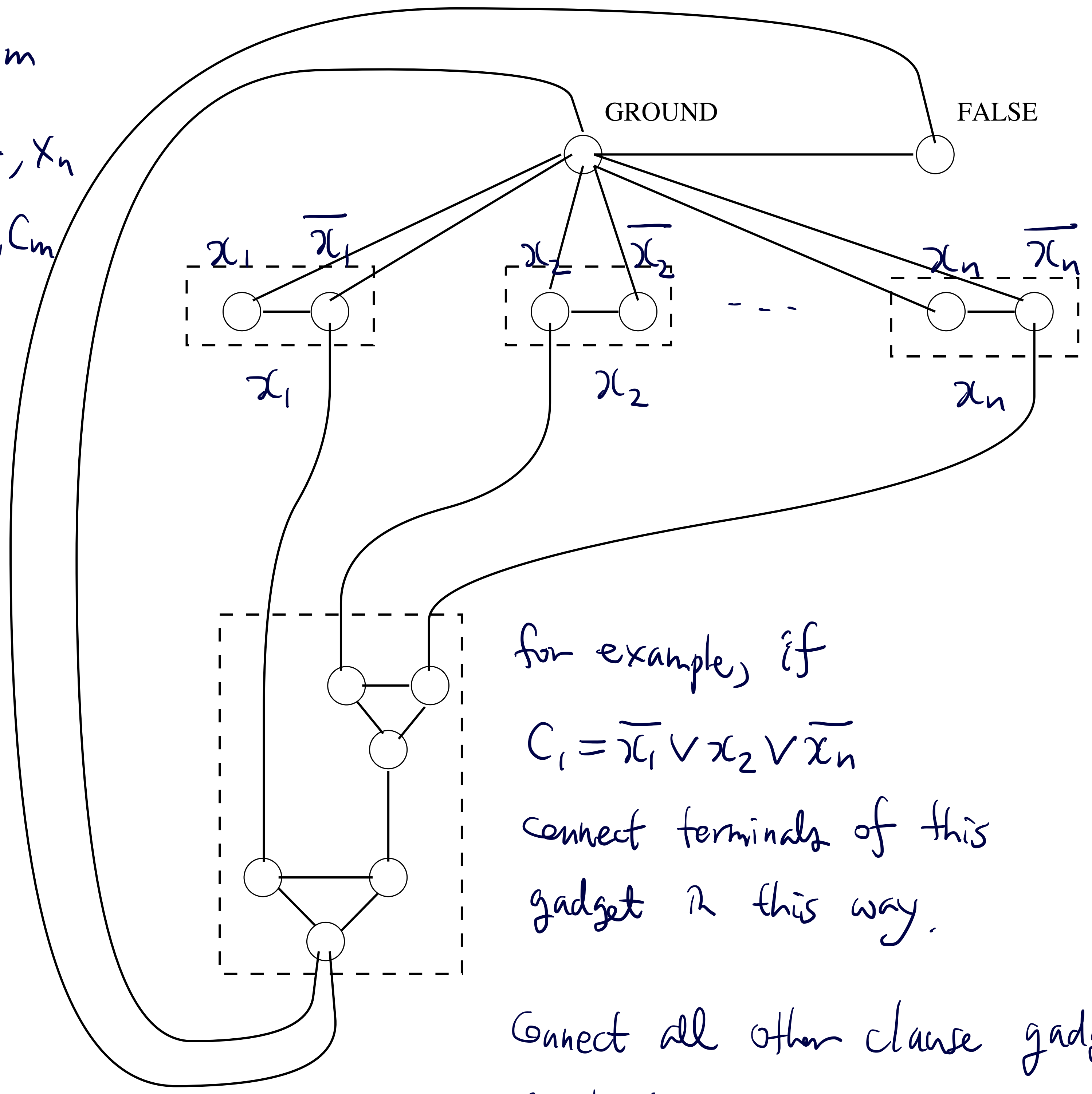


I will force M to be colored t and force T1, T2, T3 to be colored t or f.
Among $2^3=8$ possibilities for coloring T1, T2, T3, only $T_1=T_2=T_3=f$ is not possible, and other cases are all possible.

$\phi = \phi(x_1, x_2, \cdots, x_n)$

$= C_1 \wedge \cdots \wedge C_m$

$\begin{cases} n \text{ variables } x_1, \cdots, x_n \\ m \text{ clauses } C_1, \cdots, C_m \end{cases}$



the two designated verices

variable gadgets (the two nodes in one gadget correspond to literals)

clause gadgets

for example, if

$C_1 = \overline{x_1} \vee x_2 \vee \overline{x_n}$

connect terminals of this gadget in this way.

Connect all other clause gadgets similarly.

Graph colorability

- **G4C**: given a graph G, decide whether it is 4-colorable
- In general, **GnC**: given (G, n), decide whether G is n-colorable
- Theorem: **GnC** is NP-complete
  - An easy reduction from **G3C** (Send G to $(G, 3)$)

Theorem: G4C is NP-complete

- Send G to G', where G' is constructed from G by adding a new vertex and connect this new vertex with each old vertices.