CS 121: Lecture 18
Polynomial Time Reductions

Madhu Sudan

https://madhu.seas.Harvard.edu/courses/Fall2020

Book: https://introtcs.org

How to contact us
- The whole staff (faster response): CS 121 Piazza
- Only the course heads (slower): cs121.fall2020.course.heads@gmail.com

Announcements:

- Election Today!
- Midterm 2 in 2 weeks.
  - Same time format.
  - No collaboration on cheat sheets. Each person prepares their own.
  - Open book (Barak - searchable pdf).
- Sign up for active participation in Lectures.

Where we are:

Part I: Circuits:
Finite computation,
quantitative study

Part II: Automata:
Infinite restricted computation,
quantitative study

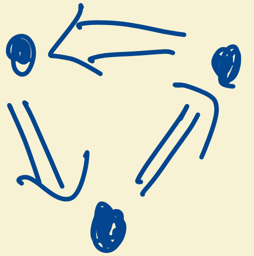
Part III: Turing Machines:
Infinite computation, qualitative study

Part IV: Efficient Computation:
Infinite computation, quantitative study

Part V: Randomized computation:
Extending studies to non-classical algorithms

Travelling Sales Problem

Coloring



3SAT

Review of last lecture

$P \subseteq EXP$

- Defined time complexity measure
- Defined classes: TIME(t(n)), P, EXP
  - Watch out for category error: All the above are sets of functions, not algorithms!
- Stated+Proved: TIME Hierarchy theorem.
  - $\Rightarrow \quad \mathrm{P} \neq \mathrm{EXP}$
- Stated:
  - RAM-TIME ≈ TM-TIME (up to polynomial factors)
  - SIZE ≤≈ TM TIME (up to polynomial factors)
  - Extended Turing-Church Thesis

Today

- Some problems in P and EXP
- Polytime Reductions $\leq_P$: Relate problems of unknown complexity
- Specific example: $\mathrm{SAT} \leq_P \ \mathrm{ISET}$

Some example problems

- Warning: Will flash lots of slides quickly!
    - Don’t have to know individual definitions/problems … just get a flavor of variety.

Solving Linear Equations

Input: n linear equations in n variables: $Ax = b$, with $A \in \mathbb{R}^{n\times n}$ and $b \in R^n$.

Output: “No solution” or assignment satisfying equations.

Notation: Equations are $\langle A_i, x\rangle = b_i$ where A_i is i-th row of A, and $\langle u, v\rangle \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} u_i v_i$

Gaussian elimination Algorithm:

- If $n = 1$: trivial
- Rearrange equations, variables, so that $A_0^0 \neq 0$ (first equation involves first variable)
- Change equation from $\langle A^0, x\rangle = b_0$ to $\frac{1}{A_0^0}\langle A_0^0, x\rangle = \frac{b_0}{A_0^0}$ to get

$$x_0 = \frac{b_0}{A_0^0} - \sum_{j=1}^{n-1} v_j x_j \quad \text{where} \quad v_j = \frac{A_j^0}{A_0^0} \qquad (*)$$

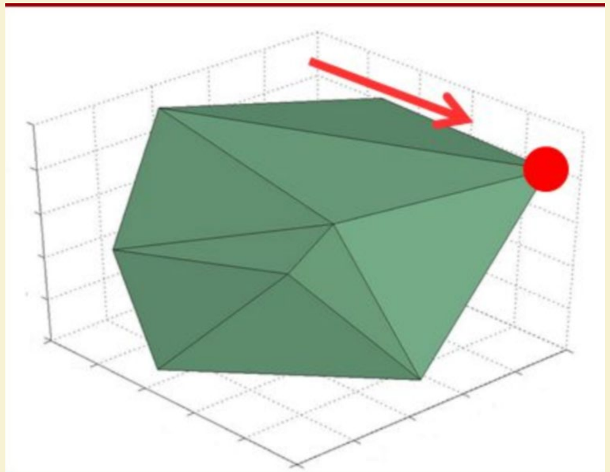
- Replace x_0 with RHS of (*) in equations $1, \dots, n-1$.
- Now have $n-1$ equations with $n-1$ variables! Repeat.

Analysis:

- Let $T(n)$ denotes number of arithmetic operations
- Then $T(n) = T(n-1) + O(n^2)$
- Yields $T(n) = O(n^3)$
- Counts #arithmetic operations. Should multiply by cost of arithmetic. (poly(#bits) per number).

- Summary: Linear equations can be solved in polynomial time.

Linear Programming



Compute $\min_{x\in\mathbb{K}} f(x)$ where f is *linear* and $\mathbb{K}$ is *polytope*:

$$\mathbb{K} = \{x \in \mathbb{R}^n \mid Ax \le b\}$$

Example: Startup resource allocation:

- *Business customer:* yields r revenue while costing a developer hours and b customer support hours.
- *End user:* yields r' revenue while costing a' dev hours and b' support hours.
- Maximize revenue subject to A total dev hours and B total support hours:
- Aside: Max, not Min! Still LP?

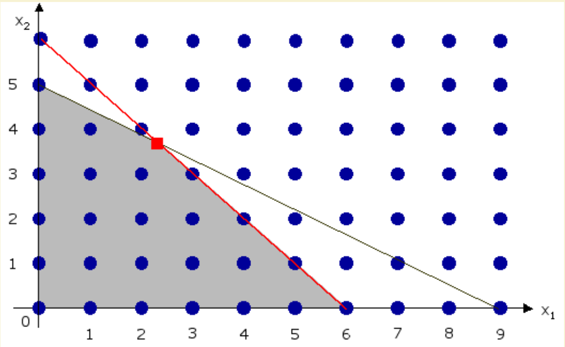
$$\begin{array}{lll} \max_{x_0,x_1\in\mathbb{R}} r\cdot x_0 & +r'\cdot x_1 & \text{s.t.} \\ a\cdot x_0 & +a'\cdot x_1 & \le A \\ b\cdot x_0 & +b'\cdot x_1 & \le B \end{array}$$

Thm: *(Khachiyan 79, Karmakar 84)* There is $poly(n)$ time algorithm for linear programming on n variables.

Integer Programming

Compute $\min_{x \in I} f(x)$ where f *linear* and $I = \mathbb{K} \cap \mathbb{Z}^n$;$\mathbb{K} = \{x \mid Ax \leq b\}$

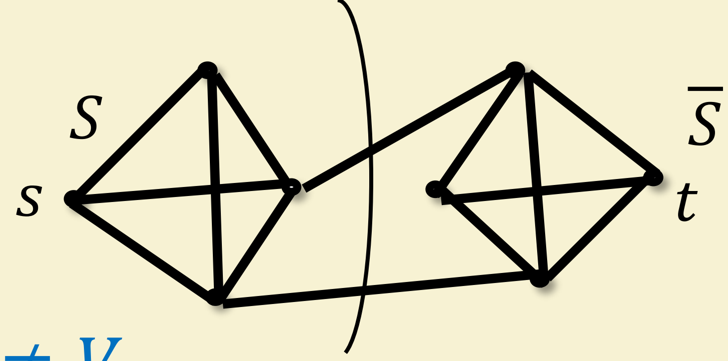
(I integer points in polytope $\mathbb{K}$)



Motivation: Can't support half a client

Depressing fact: No known polynomial time algorithm for integer programming.

Minimum s-t Cut Problem



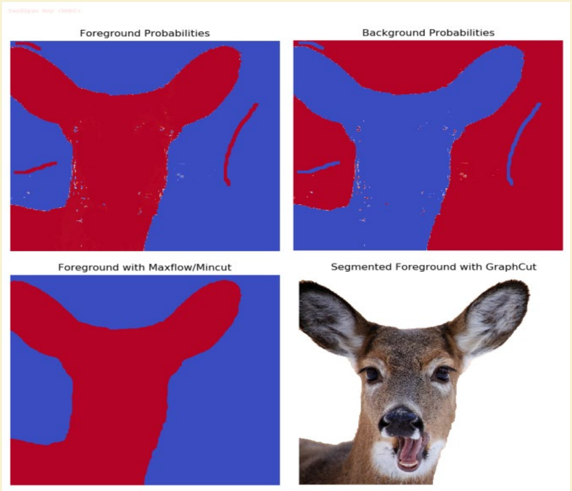
Def: A *cut* in $G = (V, E)$ is $S \subseteq V$ with $\emptyset \neq S \neq V$.

Notation: Edges cut by S, $E(S, \bar{S}) = \{(u, v) \in E \mid u \in S \Leftrightarrow v \notin S\}$

Minimum s-t cut problem:

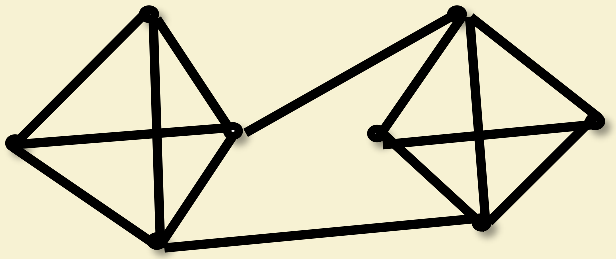
Minimize $|E(S, \bar{S})|$ over all $S \subseteq V$ s.t. $s \in S$ and $t \notin S$

(sample motivation: image segmentation)



Good News: Can be solved in time $O(VE)$

Maximum Cut Problem



Problem: Find cut S that *maximizes* $|E(S, \bar{S})|$.

Sample applications:

- Register allocation in compilers,
- Ising model, X-ray crystallography, cryo-electron microscopy, more

Depressing news: Best known algorithms are *exponential* in the worst case.

3-SAT Problem

Input: A "3CNF formula" $\phi = C_0 \wedge C_1 \wedge \cdots \wedge C_{m-1}$ on n variables $x_0 \ldots x_{n-1}$

3CNF Formula = AND of m 3-clauses $(\phi = C_0 \wedge C_1 \wedge \cdots \wedge C_{m-1})$

3-Clause = OR of 3 literals $(C_j = \ell_1 \vee \ell_2 \vee \ell_3)$

literal = variable or its negation $(\ell = x_i \quad \text{or} \quad \ell = \overline{x_i})$

Goal: Output 1 if there is assignment $x \in \{0,1\}^n$ that makes formula *true*.

Output 0 otherwise. AND

OR

Example: $(x_7 \vee \overline{x}_{17} \vee x_{29}) \wedge (\overline{x}_7 \vee x_{15} \vee x_{22}) \wedge (x_{22} \vee \overline{x}_{29} \vee x_{55})$

$$(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x_3})$$
$$\wedge (x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

3-SAT Problem

Input: A "3CNF formula" $\phi = C_0 \wedge C_1 \wedge \cdots \wedge C_{m-1}$ on n variables $x_0 \ldots x_{n-1}$

Goal: Determine if there is assignment $x \in \{0,1\}^n$ that makes formula *true*.

Depressing news: Best known algorithms require exponential time.

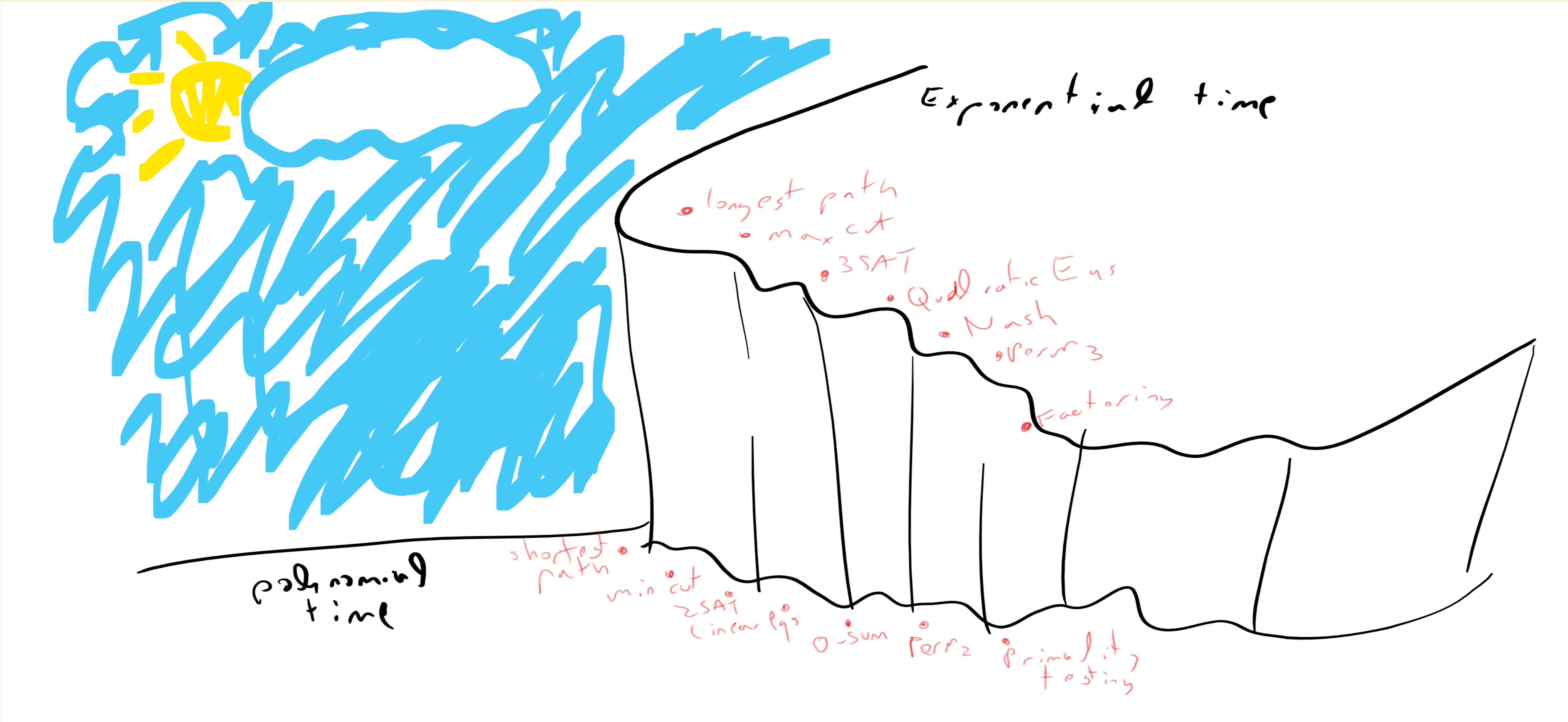
Better news: Exponent has improved, decent heuristic "SAT SOLVERS", *can* solve the 2SAT problem.

Exercise: Show "INTEGER PROGRAMMING can be solved in poly time $\Rightarrow$ 3SAT can be solved in poly time".

Summary of problems:

- The following problems are in $\mathbf{P}$
  - Does the **linear system** $Ax = b$ have a solution?
  - Does the **linear program** $Max\ \langle c, x\rangle\ s.t.\ Ax \le b$ have a real solution of value $\ge v$
  - Given G does G have a cut of value at most k (**min cut**)
  - Given a **2CNF** formula $\phi = C_0 \wedge C_1 \wedge \cdots \wedge C_{m-1}$ is there a satisfying $x \in \{0,1\}^n$
- The following problems are in $\mathbf{EXP}$ but not known to be in $\mathbf{P}$
  - Does **linear program** $Max\ \langle c, x\rangle\ s.t.\ Ax \le b$ have an **integer** solution of value $\ge v$
  - Given G does G have a cut of value at least k (**max cut**)
  - Given a **3CNF** formula $\phi = C_0 \wedge C_1 \wedge \cdots \wedge C_{m-1}$ is there a satisfying $x \in \{0,1\}^n$
- HALT is not in $\mathbf{P}$ or even $\mathbf{EXP}$!
- HALT_{2^n} is in **EXP** but not **P**.

Artwork representing lecture thus far!



Relating problems to each other?

- EXP seems to have many interesting problems we would like to solve.
- Questions: Which ones are in P?
  - Answer: "Don't know!"

- If we can't determine the answer to any of these questions, can we at least relate the answers?
  - Answer: Yes!
  - Tool: (Polynomial time) Reductions!

Maximum Cut as Integer Program

FLASH

Input: $G = (V, E)$ with n vertices and m edges
Goal: Find cut $\emptyset \neq S \neq V$ maximizing $|E(S, \overline{S})|$.

IP formulation: Find $x \in \mathbb{Z}^n, y \in \mathbb{Z}^m$ maximizing $\sum_{j=0}^{m-1} y_j$ s.t.

$0 \leq x_i \leq 1$ for $i \in [n]$ (*i.e.*, $x \in \{0,1\}^n$)

$0 \leq y_j \leq 1$ for $j \in [m]$ ($y \in \{0,1\}^m$)

For jth edge (u, v):

$$y_j \leq x_u + x_v \quad (x_u = x_v = 0 \Rightarrow y_j = 0)$$

$$y_j \leq (1 - x_u) + (1 - x_v) \quad (x_u = x_v = 1 \Rightarrow y_j = 0)$$

Reductions:

- Previous page was a reduction from X to Y.
  - X = ?, Y = ?
  - Consequence:

MAX CUT $\leq_p$ IP

Based on what we've shown so far $\Rightarrow$ Possible

MAX-CUT \ IP	In P?	Not in P?
In P?	Possible	2
Not in P?	1 Not Possible!	Possible

Truth (though we didn't show it): 2 is also not possible

Can reduce IP $\leq_p$ MAX CUT

Polynomial Time Reductions

computable

DEFINITION:

poly time

- For Boolean functions $F, G: \{0,1\}^* \to \{0,1\}$, $F \leq_P G$ if there is a ~~polynomial time algorithm~~ function $R: \{0,1\}^* \to \{0,1\}^*$ s.t. for every $x \in \{0,1\}^*$,

$$F(x) = 1 \Leftrightarrow G(R(x)) = 1$$

$$F(x) = G(R(x))$$

- In the example before:
  - $F(H,k) = MAXCUT(H,k) = 1 \Leftrightarrow H\ has\ a\ cut\ of\ size \geq k$
  - $G(A,b,c,v) = IP(A,b,c,v) = 1 \Leftrightarrow \exists x \in \mathbb{Z}^n$ s.t. $c.x \geq v$ and $Ax \leq b$
  - Reduction $R(H,k) = (A,b,c,v)$ … (all the constraints and objective).
  - Worked because:
    - R polytime computable!
    - $IP(A,b,c,v) = MaxCut(H,k)$

R cannot be exp. time computable.

$\boxed{\text{Example 2}}$ 3-$SAT \leq_P IP$:

FLASH

Input: A 3CNF formula $C_0 \wedge C_1 \wedge \cdots \wedge C_{m-1}$ on n variables (where C_j's are 3-clauses)

Goal: Find out if there is assignment $x \in \{0,1\}^n$ that makes formula *true*.

IP formulation: Maximize over $x \in \mathbb{Z}^n, y \in \mathbb{Z}^m$, the quantity $\sum_{j=0}^{m-1} y_j$ subject to:

$0 \leq x_i \leq 1$, $0 \leq y_j \leq 1$

For jth clause, say $C_j = (\overline{x}_{17} \vee x_{55} \vee x_{22})$:

($x_{17} = 1$ and $x_{55} = 0$ and $x_{22} = 0 \Rightarrow y_j = 0$)

$y_j \leq (1 - x_{17}) + x_{55} + x_{22}$

Claim: ϕ satisfiable $\Leftrightarrow \sum y_j = m$

Surprising fact: (Converse is also true) If you can solve 3SAT in poly time then you can solve INTEGER PROGRAMMING in poly time!

Independent Set (ISET):

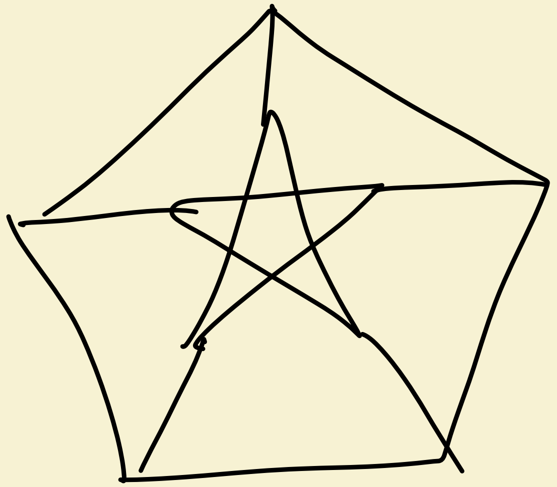
3SAT $\leq_p$ ISET

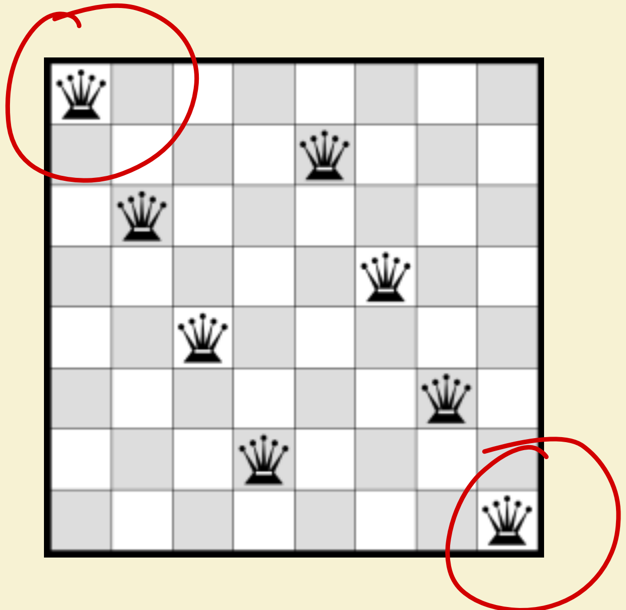
Input: $(G = (V, E), k)$

Question: Does there exists an independent set of size $\geq k$ in G

where $S \subseteq V$ is independent if no edges within S

$(\forall u, v \in S, \ (u, v) \notin E \ )$





3SAT $\leq_p$ ISET

Theorem: Suppose that $ISET \in \boldsymbol{P}$ then $3SAT \in \boldsymbol{P}$

Corollary: Suppose that $3SAT \notin \boldsymbol{P}$ then $ISET \notin \boldsymbol{P}$

Why?

Proof: We will show poly-time $R: \{0,1\}^* \to \{0,1\}^*$

$R: \{3CNF\ formulas\} \to \{\ (graphs, numbers)\}$

s.t. for every 3CNF φ , $3SAT(\varphi) = ISET(R(\varphi))$

Q: Why is this enough?

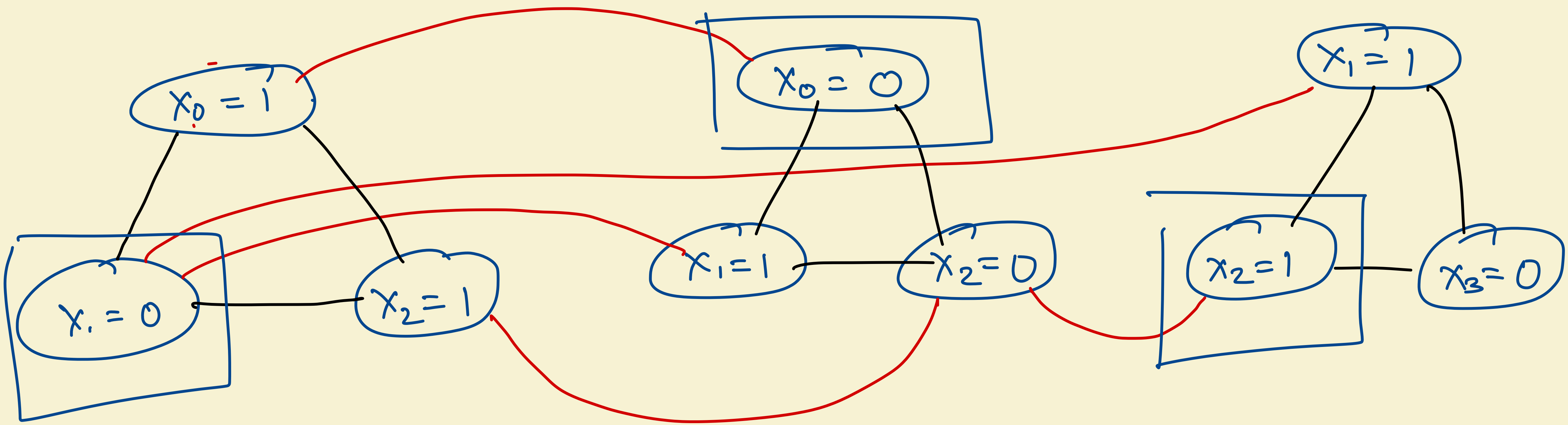
Theorem: Suppose that $ISET \in \boldsymbol{P}$ then $3SAT \in \boldsymbol{P}$ $3SAT(\phi)=1$

Proof: We will show poly-time $R: \{3CNF\ formulas\} \to \{\ (graphs, numbers)\}$

s.t. for every 3CNF φ , $3SAT(\varphi) = ISET(R(\varphi))$ $\Leftarrow$ G has ind. set of size m

Example: $\varphi = (x_0 \vee \overline{x_1} \vee x_2) \wedge (\overline{x_0} \vee x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_2 \vee \overline{x_3})$

(has $\mathrm{n} = 4$ variables, $\mathrm{m} = 3$ clauses)



Theorem: Suppose that $ISET \in \mathbf{P}$ then $3SAT \in \mathbf{P}$

Proof: We will show poly-time $R: \{3CNF\ formulas\} \to \{\ (graphs, numbers)\}$ s.t. for every 3CNF φ , $3SAT(\varphi) = ISET(R(\varphi))$

Algorithm R:

Input: φ – 3CNF with n variables and m clauses

1. Let G be graph with $3m$ vertices we will name $(j,1), (j,2), (j,3)$ for $j \in [m]$
2. For every $j \in [m]$, add edges $(j,1) - (j,2)$, $(j,2) - (j,3)$, $(j,3) - (j,1)$
3. For every pair of clauses C_j and $C_{j'}$ if literals (j,a) and (j',b) conflict (one negation of the other) then add the edge $(j,a) - (j',b)$
4. Return (G, m)

Claim 1 (completeness): If $3SAT(\varphi) = 1$ then $ISET(G,m) = 1$.

Claim 2 (soundness): If $ISET(G,m) = 1$ then $3SAT(\varphi) = 1$. $\Longleftrightarrow$ if $3SAT(\phi)=0$ then $ISET(\ )=0$

Existing system of Logic

New One

complete

$\tilde{T}$ is true

T is true

soundness

$\tilde{T}$ is false

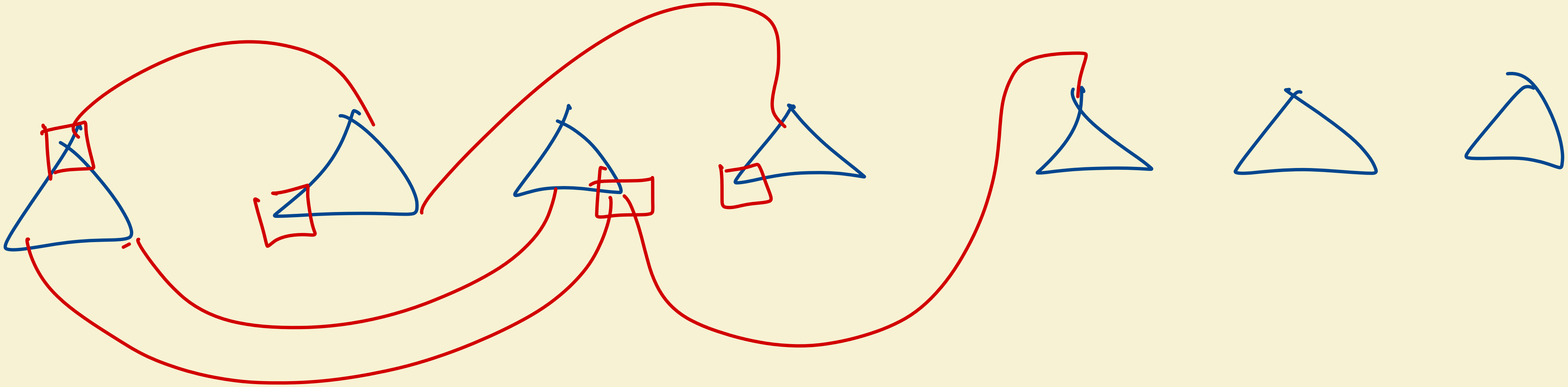
T is false

$\sim \Rightarrow R$

$$F(x) = 1 \xrightarrow[\text{complete}]{} G(R(x)) = 1$$

$$F(x) = 0 \xrightarrow[\text{soundness}]{} G(R(x)) = 0$$

Given $S \subseteq$, S indepent, $|S| = m$



$\Downarrow$ can get assignment to $x_0 = \_$ $x_1 = \_$ $x_2 = \ldots$ $x_{n-1} = \_$

s.t. every clause is satisfied.

$$x_{i_1} \vee \overline{x_{i_2}} \vee x_{i_3}$$

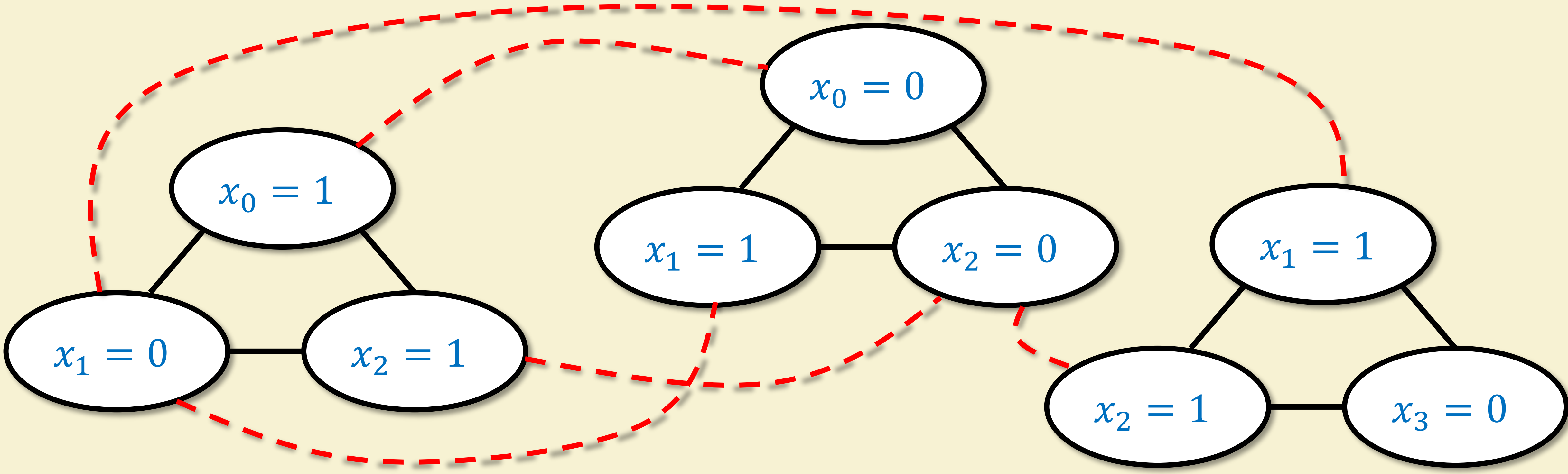
Theorem: Suppose that $ISET \in \boldsymbol{P}$ then $3SAT \in \boldsymbol{P}$

Proof: We will show poly-time $R: \{3CNF\ formulas\} \to \{\ (graphs, numbers)\}$

s.t. for every 3CNF φ , $3SAT(\varphi) = ISET(R(\varphi))$

Example: $\varphi = \ (x_0 \vee \overline{x_1} \vee x_2) \wedge (\overline{x_0} \vee x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_2 \vee \overline{x_3})$

(has $\mathrm{n} = 4$ variables, $\mathrm{m} = 3$ clauses)



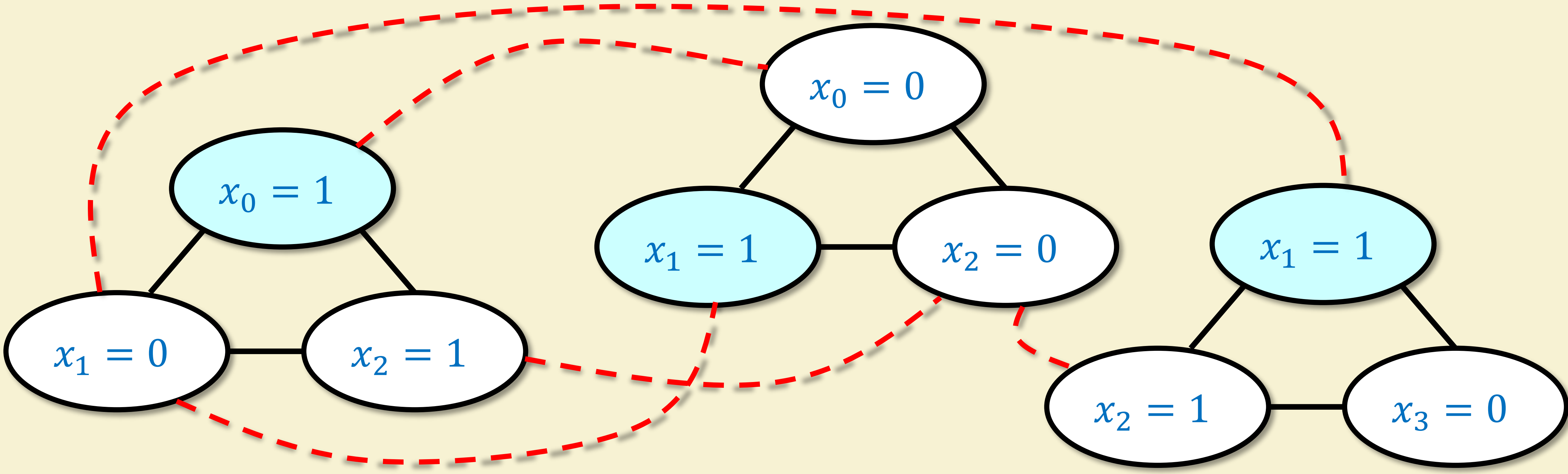
Theorem: Suppose that $ISET \in \boldsymbol{P}$ then $3SAT \in \boldsymbol{P}$

Proof: We will show poly-time $R: \{3CNF\ formulas\} \to \{\ (graphs, numbers)\}$

s.t. for every 3CNF φ , $3SAT(\varphi) = ISET(R(\varphi))$

Example: $\varphi = \ (x_0 \vee \overline{x_1} \vee x_2) \wedge (\overline{x_0} \vee x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_2 \vee \overline{x_3})$

(has $\mathrm{n} = 4$ variables, $\mathrm{m} = 3$ clauses)



$x = 1101$

Algorithm R:

Input: φ – 3CNF with n variables and m clauses

1. Let G be graph with $3m$ vertices we will name $(j,1),(j,2),(j,3)$ for $j \in [m]$
2. For every $j \in [m]$, add edges $(j,1)-(j,2)$, $(j,2)-(j,3)$, $(j,3)-(j,1)$
3. For every pair of clauses C_j and $C_{j'}$ if literals (j,a) and (j',b) conflict (one negation of the other) then add the edge $(j,a)-(j',b)$
4. Return (G,m)

Claim 1 (completeness): If $3SAT(\varphi)=1$ then $ISET(G,m)=1$.

Proof: Assume x satisfies φ. Then for every clause C_j there is a literal (j,a) satisfied.

add (j,a) to set S. The size of S is m.

We claim that S is an independent set:

- S contains one vertex in each triangle so no "black" edges.
- If vertex tagged as "$x_i = 0$" in S then "$x_i = 1$" can't be in S so no "red" edges.

$\blacksquare$

Algorithm R:

Input: φ – 3CNF with n variables and m clauses

1. Let G be graph with $3m$ vertices we will name $(j,1),(j,2),(j,3)$ for $j \in [m]$
2. For every $j \in [m]$, add edges $(j,1)-(j,2)$, $(j,2)-(j,3)$, $(j,3)-(j,1)$
3. For every pair of clauses C_j and $C_{j'}$ if literals (j,a) and (j',b) conflict (one negation of the other) then add the edge $(j,a)-(j',b)$
4. Return (G,m)

Claim 2 (soundness): If $ISET(G,m)=1$ then $3SAT(\varphi)=1$.

Proof: Assume S independent set of size m in G.

Q: Show that S contains exactly one vertex per triangle   Hint:

Set $x_i^*=1$ if S contains vertex tagged "$x_i=1$" otherwise $x_i=0$.

For every clause C_j there is vertex in S tagged "$x_i=b$". We claim $x_i^*=b$

If $b=1$: by definition.

If $b=0$: S can't contain vertex tagged "$x_i=1$" since it's independent.

Polynomial-time reductions

$3SAT \leq_p ISET$

We showed: Poly time A for $ISET$ $\Rightarrow$ Poly time B for $3SAT$

φ

Alg B

$G, k = R(\varphi)$

Alg A

$ISET(G,k) = 3SAT(\varphi)$

Def: Let $F, G: \{0,1\}^* \to \{0,1\}$. We say $F \leq_p G$ if $\exists$ poly-time $R: \{0,1\}^* \to \{0,1\}^*$ s.t. $\forall_{x \in \{0,1\}^*} F(x) = G(R(x))$

Q: Prove that if $F \leq_p G$ and $G \leq_p H$ then $F \leq_p H$

Next Lecture:

More systematic exploration of the “hard” problems from today.

All share a common feature: What?

Leads us to NP and NP-completeness!