Slides for CIS 675

NP-Complete Problems, Part 2

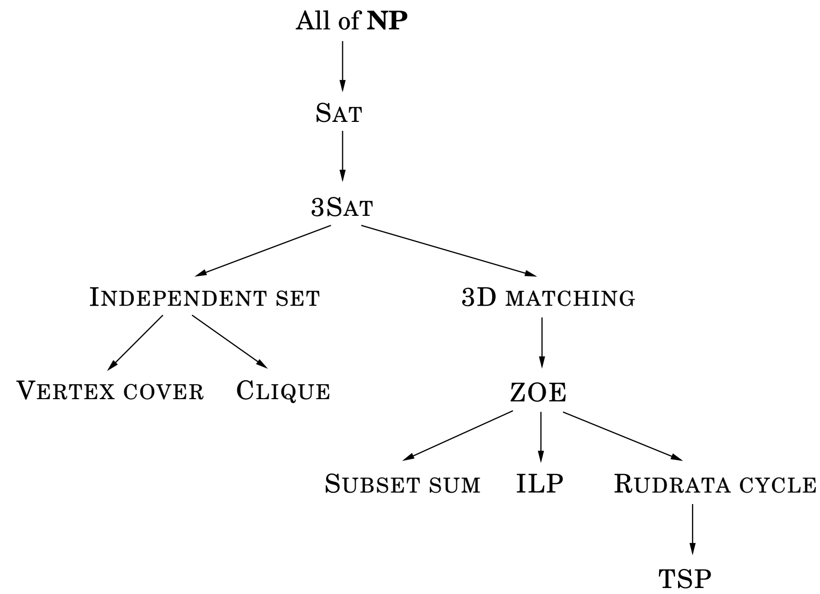
Jim Royer

DPV Chapter 8

April 3, 2019

Uncredited diagrams are from DPV or homemade.

The Plan of §8.3



Warm up: Rudrata (s,t)-Path $\trianglelefteq$ Rudrata Cycle

Rudrata (s,t)-Path

Given: $G = (V, E)$ and $s, t \in V$.
Find: A path from s to t in G passing through each vertex once.

Rudrata Cycle

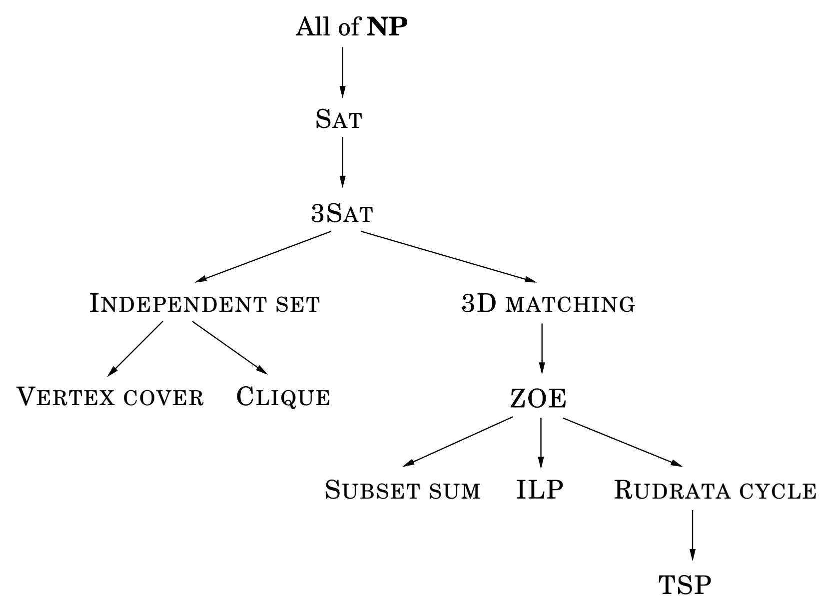
Given: $G = (V, E)$.
Find: A cycle in G passing through each vertex once.

Rudrata (s,t)-Path $\trianglelefteq$ Rudrata Cycle:

- Given an instance of Rudrata (s,t)-Path, (G,s,t), construct G' by adding an new vertex x and new edges (x,s) and (x,t).
- If P is a Rudrata-cyclc in G', then leaving x, (x,s) and (x,t) out of P gives an (s,t)-Rudrata path in G.
- If G has a (s,t)-Rudrata path P, then adding (x,s) and (x,t) to P yields a Rudrata cycle in G'.

$\therefore$ If G' has no Rudrata cycle, then G has no (s,t)-Rudrata path.

The Plan of §8.3



3SAT $\trianglelefteq$ Independent Set, 1

3SAT

Given: A CNF formula θ in which each clause has at most 3 literals.
Find: A satisfying assignment for θ.

Independent Set Problem

Given: $G = (V, E)$ and b.
Find: An independent set for G of size $\geq b$.
I.e., Find $U \subseteq V$ with $|U| \geq b$ and $(\forall u, v \in U)[(u, v) \notin E]$.

Puzzle:
These are very different looking problems.
How to we get a reduction?

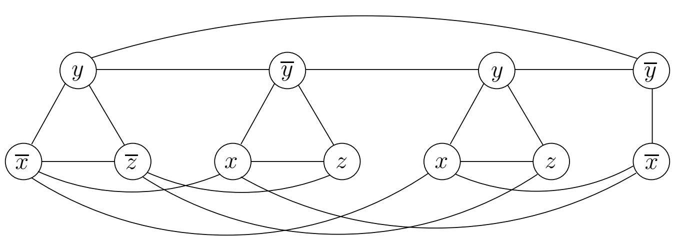
Think circuits, but not too literally.

3SAT $\trianglelefteq$ Independent Set, 2

- To build a satisfying assignment of an instance of 3SAT, we have to pick out at least one literal per clause to be TRUE — consistently!

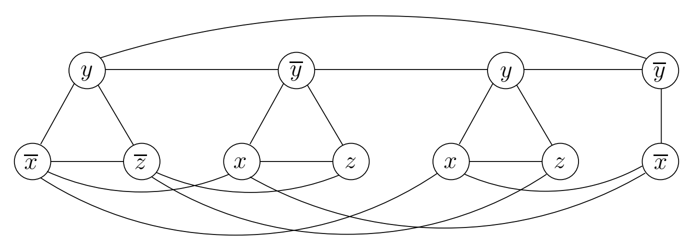
3SAT $\trianglelefteq$ Independent Set, 2

- To build a satisfying assignment of an instance of 3SAT, we have to pick out at least one literal per clause to be TRUE — consistently!
  Consider: $(\overline{x} \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y})$.



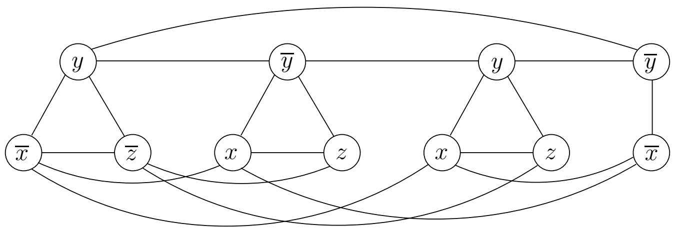
3SAT $\trianglelefteq$ Independent Set, 2

- To build a satisfying assignment of an instance of 3SAT, we have to pick out at least one literal per clause to be TRUE — consistently!
  Consider: $(\overline{x} \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y})$.
- A clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ is represented by a triangle with verts labeled: ℓ_1, ℓ_2, & ℓ_3.



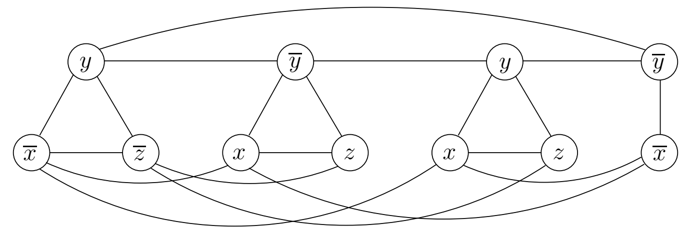
3SAT $\trianglelefteq$ Independent Set, 2

- To build a satisfying assignment of an instance of 3SAT, we have to pick out at least one literal per clause to be TRUE — consistently!
  Consider: $(\overline{x} \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y})$.
- A clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ is represented by a triangle with verts labeled: ℓ_1, ℓ_2, & ℓ_3.
- *Idea:* If $v \in U$ an indep. set and v has label ℓ, then the truth assignment corresponding to U sets ℓ to TRUE.



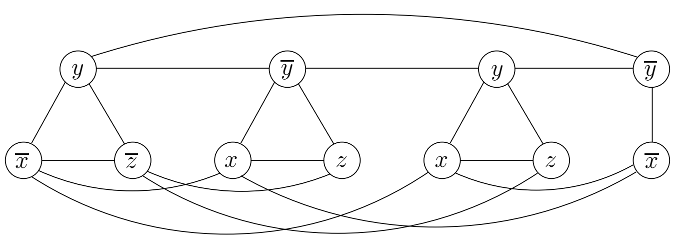
3SAT $\trianglelefteq$ Independent Set, 2

- To build a satisfying assignment of an instance of 3SAT, we have to pick out at least one literal per clause to be TRUE — consistently!
  Consider: $(\overline{x} \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y})$.
- A clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ is represented by a triangle with verts labeled: ℓ_1, ℓ_2, & ℓ_3.
- *Idea:* If $v \in U$ an indep. set and v has label ℓ, then the truth assignment corresponding to U sets ℓ to TRUE.
- Set g = # of clauses. So any indep. set must have one vertex from each triangle. *(Why?)*



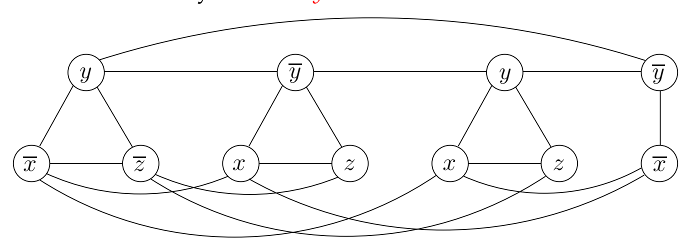
3SAT $\trianglelefteq$ Independent Set, 2

- To build a satisfying assignment of an instance of 3SAT, we have to pick out at least one literal per clause to be TRUE — consistently!
  Consider: $(\overline{x} \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y})$.
- A clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ is represented by a triangle with verts labeled: ℓ_1, ℓ_2, & ℓ_3.
- *Idea:* If $v \in U$ an indep. set and v has label ℓ, then the truth assignment corresponding to U sets ℓ to TRUE.
- Set g = # of clauses. So any indep. set must have one vertex from each triangle. *(Why?)*
- But we have to have consistency checks. We don't want to set x and $\overline{x}$ to TRUE.



3SAT $\trianglelefteq$ Independent Set, 2

- To build a satisfying assignment of an instance of 3SAT, we have to pick out at least one literal per clause to be TRUE — consistently!
  Consider: $(\overline{x} \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y})$.
- A clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ is represented by a triangle with verts labeled: ℓ_1, ℓ_2, & ℓ_3.
- *Idea:* If $v \in U$ an indep. set and v has label ℓ, then the truth assignment corresponding to U sets ℓ to TRUE.
- Set g = # of clauses. So any indep. set must have one vertex from each triangle. *(Why?)*
- But we have to have consistency checks. We don't want to set x and $\overline{x}$ to TRUE.
- So put an edge between each occurrence of a variable x and each occurrence of $\overline{x}$. This enforces consistency. *(Why?)*



3SAT ⊴ Independent Set, 3

Construction: 3SAT instance ↦ IS instance

- Given an instance of 3SAT $C_1, \ldots, C_k$.
- For each clause $C_i = (\ell_1, \ell_2, \ell_3)$, build a triangle with vertices labeled ℓ_1, ℓ_2, & ℓ_3.
- Add an edge between each literal and all of its opposites.
- Set $g = k$.

Construction: IS solution ↦ 3SAT solution

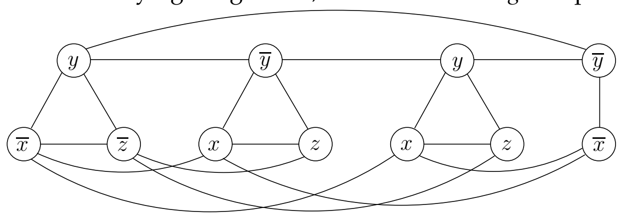
- Given an independent set U: for each $v \in U$ with label ℓ, set ℓ to TRUE.
- For each variable that does not yet have a truth value, set it to TRUE.

Claim 1: Both constructions are poly-time.

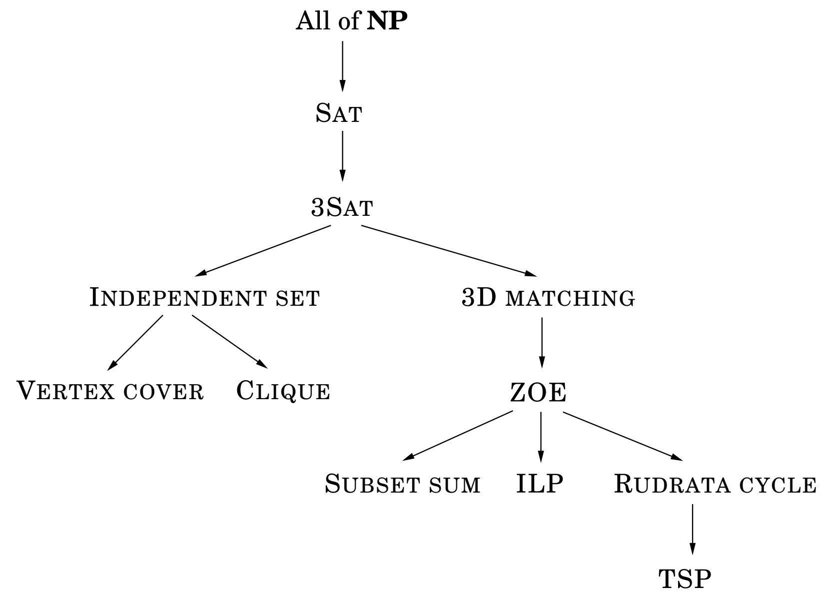
Claim 2: Given an indep. set, we can construct a satisfying assignment for $C_1, \ldots, C_k$.

Claim 3′: If $C_1, \ldots, C_k$ has an satisfying assignment, then there is size g indep. set.

$(\overline{x} \vee y \vee \overline{z}) \wedge$
$(x \vee \overline{y} \vee z) \wedge$
$(x \vee y \vee z) \wedge$
$(\overline{x} \vee \overline{y})$



The Plan of §8.3



SAT ⊴ 3SAT

Construction: SAT instance ↦ 3SAT instance

- Translate each clause to sequence of clauses.
- $(\ell_1 \vee \ell_2 \vee \cdots \vee \ell_k) \longmapsto (\ell_1 \vee \ell_2 \vee \mathbf{y}_1)\ (\overline{\mathbf{y}_1} \vee \ell_3 \vee \mathbf{y}_2) \cdots (\overline{\mathbf{y}_{k-3}} \vee \ell_{k-1} \vee \ell_k)$ where $k > 3$ and $\mathbf{y}_1, \ldots, \mathbf{y}_{k-3}$ are new vars.
- Clauses with ≤ 3 literals translate to themselves.

Construction: 3SAT solution ↦ SAT solution

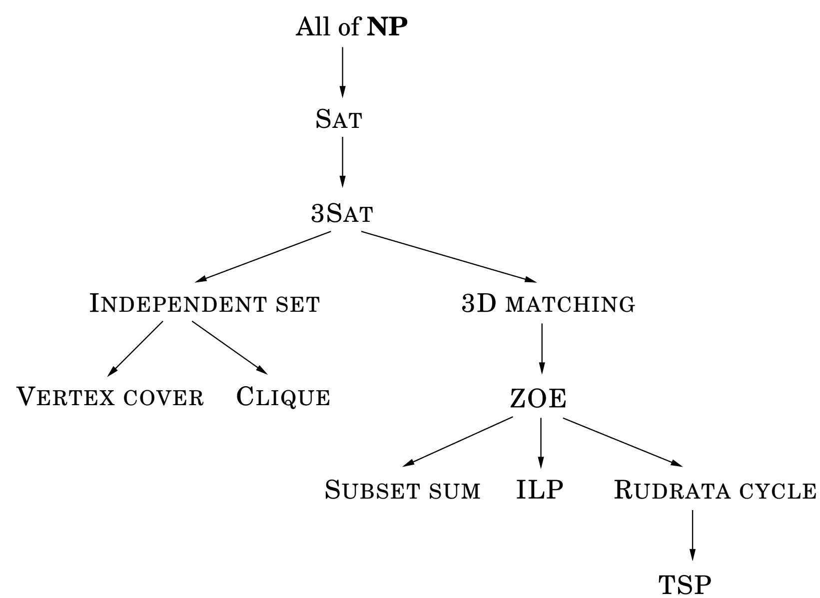
- Take the truth assignment for the 3SAT formula & restrict it to the original vars.

Claim 1: Both constructions are poly-time.

Claim 2: If the 3SAT formula has satisfying assignment, then restriction satisfies the SAT formula.

Claim 3′: If the SAT formula is satisfiable, so is the 3SAT formula.

The Plan of §8.3

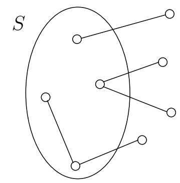


Independent-Set $\trianglelefteq$ Vertex Cover

Definition

Suppose $G = (V, E)$ is an undirected graph and $S \subseteq V$.

- (a) S is *independent* when for each $u, v \in S$, $(u, v) \notin E$.
- (b) S is a *vertex cover* when each edge of E has at least one endpoint in S.



Independent Set Problem

Given: G and b.
Find: An indep. set for G of size $\geq b$.

Vertex Cover Problem

Given: G and b.
Find: A vertex cover for G of size $\leq b$.

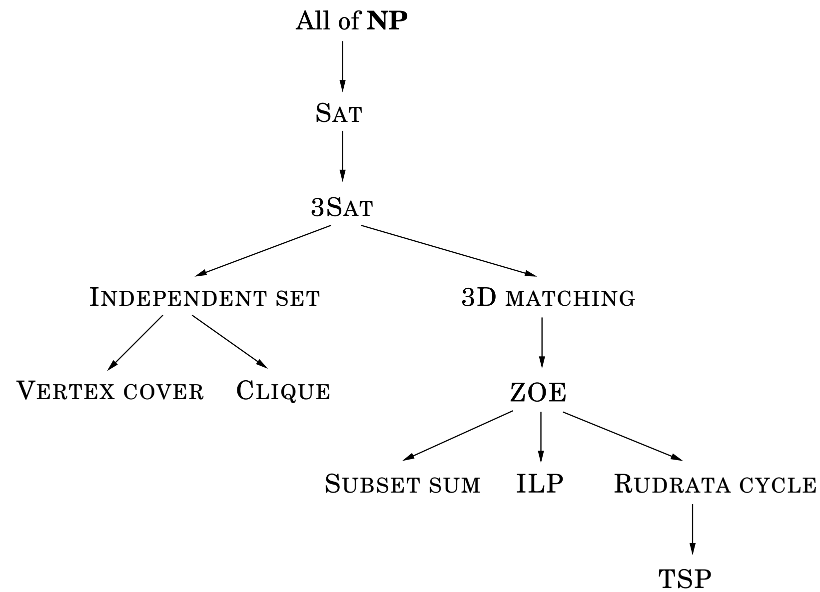
Reduction:

$$f((V, E), b) = ((V, E), |V| - b).$$
$$h(S) = V - S.$$

Claims

1. Both constructions are poly-time.
2. If U is a vertex cover for G with $|U| \leq |V| - b$, then $V - U$ is an indep. set with $|V - U| \geq b$.
3. If G has a indep. set of size $\geq b$, then G has a vertex cover of size $|V| - b$.

The Plan of §8.3



Independent-Set $\trianglelefteq$ Clique

Definition

Suppose $G = (V, E)$ is an undirected graph and $S \subseteq V$.

- (a) S is *independent* when for each $u, v \in S$, $(u, v) \notin E$.
- (b) U is a *clique* when for each distinct $u, v \in U$, $(u, v) \in E$.
- (c) $\overline{G} = (V, \overline{E})$ where $\overline{E} = \{ (u, v) : (u, v) \notin E \}$.

Lemma

S is an independent set in G $\iff$ S is a clique in $\overline{G}$.

Independent Set Problem

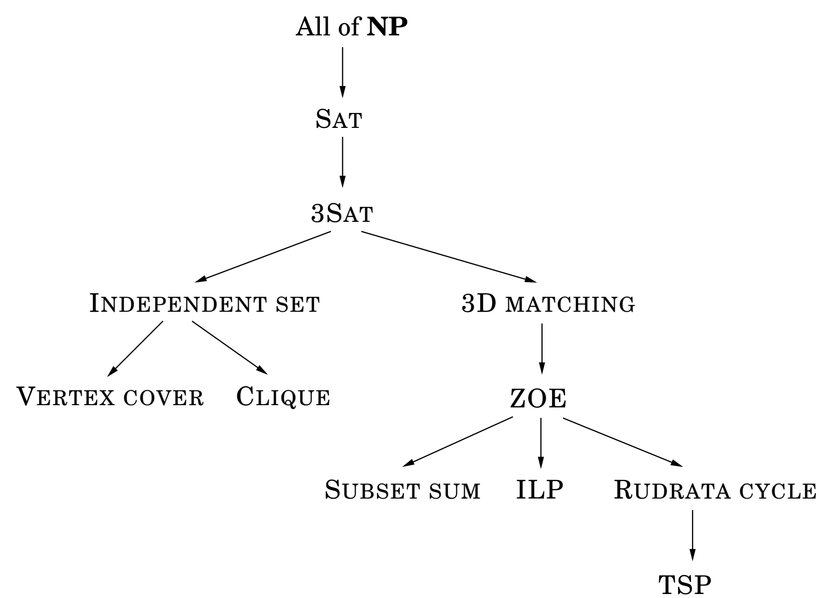
Given: G and b.
Find: An indep. set for G of size $\geq b$.

Clique Problem

Given: G and b.
Find: A clique in G of size $\geq b$.

$$f(\,(V, E), b\,) = ??$$
$$h(S) = ??$$

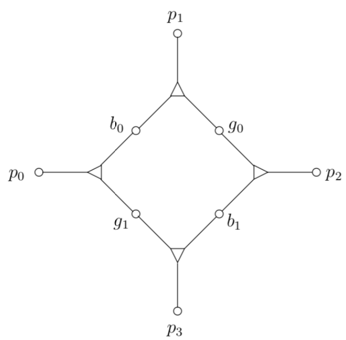
The Plan of §8.3



3SAT $\trianglelefteq$ 3D Matching, 1

3D Matching

Given: $R \subseteq B \times G \times P$ with $|B| = |G| = |P| = n$.
Find: A size-n subset $M \subseteq R$ such that
if (b,g,p) and (b',g',p') are distinct elements of M,
then $b \neq b'$, $g \neq g'$, and $p \neq p'$.
Notes: B = boys, G = girls, P = pets.
Exactly one of each Boy, Girl, Pet is chosen in the matching.



- Four triples:
  (b_0,g_1,p_0), (b_1,g_0,p_2), (b_0,g_0,p_1), (b_1,g_1,p_3)
- b_0, b_1, g_0, g_1 appear only in these triples.
- Any 3DM has to take either
  - (b_0,g_1,p_0), (b_1,g_0,p_2) can represent True
  - (b_0,g_0,p_1), (b_1,g_1,p_3) can represent False
- For each variable (from the 3SAT instance) x, create a copy of this gadget where b_0^x, $b_1^x \in B$, g_0^x, $g_1^x \in G$, and $p_0^x, p_1^x \in P$

3SAT $\trianglelefteq$ 3D Matching, 2

Assumption: Preprocessing the 3SAT Instance

Without loss of generality, we assume no literal in the 3SAT instance has more than two occurrences.

- For a clause $c = (x \vee \overline{y} \vee z)$ add $b_c \in B$ and $g_c \in G$ and add triples
  - (b_c,g_c,p_1^x) (or else (b_c,g_c,p_3^x)) (if x is true, then p_1^x and p_3^x are available)
  - (b_c,g_c,p_0^y) (or else (b_c,g_c,p_2^y)) (if y is false, then p_0^y and p_2^y are available)
  - (b_c,g_c,p_1^z) (or else (b_c,g_c,p_3^z)) (if z is true, then p_1^z and p_3^z are available)
- Similarly, for the other clauses.
- Make sure each literal in clause has has a different p_i to match with b_c and g_c. (Here is where the above assumption comes in.)

Clean-up:

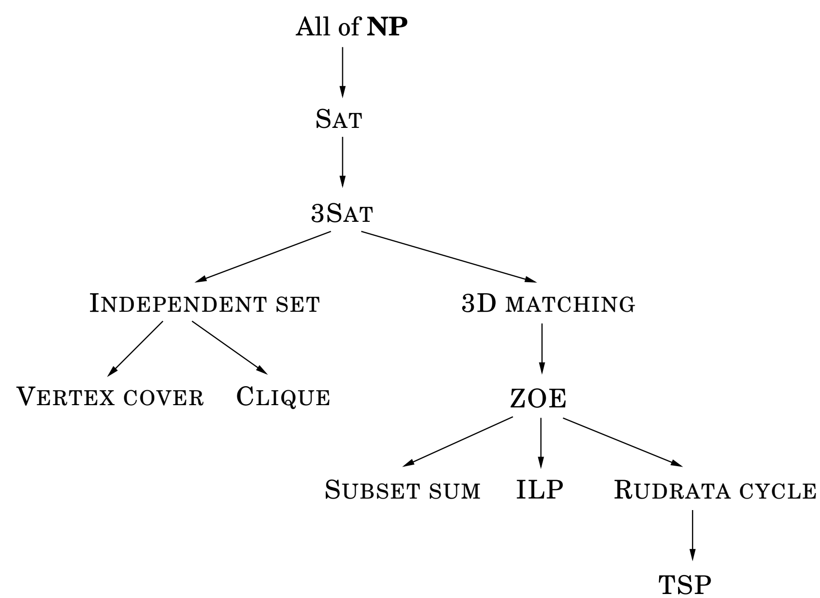
- Suppose the 3SAT instance has n variables and m clauses.
- For $j = 1, \ldots, 2n - m$, add $b_i^* \in B$, $g_i^* \in G$ and the set of triples $\{\, (b_i^*, g_i^*, p_j^v) \;:\; v \in V,\ j = 0, \ldots, 3 \,\}$. I.e., b_i^* and g_i^* like any P.

3SAT $\trianglelefteq$ 3D Matching, 3

Claim

- The construction is poly-time.
- If the instance of 3SAT is satisfiable, then there is a 3D matching in the set of triples constructed.
- If there is a 3D matching in the set of triples constructed, then the instance of 3SAT is satisfiable.

The Plan of §8.3



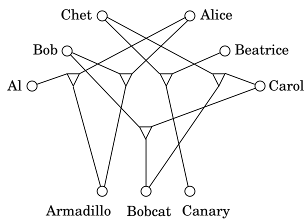
3D Matching $\trianglelefteq$ ZOE, 1

3D Matching

Given: $R \subseteq U \times V \times W$ with $|U| = |V| = |W| = n$.
Find: A size-n subset $M \subseteq R$ such that
if (u, v, w) and (u', v', w') are distinct elements of M, *then* $u \neq u'$, $v \neq v'$, and $w \neq w'$.

Zero-One Equations (ZOE)

Given: A, an $m \times n$ matrix of 0's and 1's
Find: $\vec{x}$, a n-vector of 0's and 1's such that $A\vec{x} = \vec{1}$.



$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Construction

- Suppose $m = |R|$.
- x_j variable for the j-th triple.
- $x_j = 1 \iff j$-th triple is used.
- $(A_{i,j})$ is $3m \times n$
- $A_{i,j} = 1 \iff$ the j-th elm. of U is used in the i-th triple.
- $A_{m+i,j} = 1 \iff$ the i-th elm. of V is used in the j-th triple.
- $A_{2m+i,j} = 1 \iff$ the i-th elm. of W is used in the j-th triple.
- So $A\vec{x} = \vec{1}$ means ...

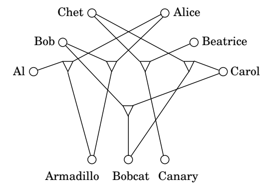
3D Matching $\trianglelefteq$ ZOE, 2

3D Matching

Given: $R \subseteq U \times V \times W$ with $|U| = |V| = |W| = n$.
Find: A size-n subset $M \subseteq R$ such that
if (u, v, w) and (u', v', w') are distinct elements of M, *then* $u \neq u'$, $v \neq v'$, and $w \neq w'$.

Zero-One Equations (ZOE)

Given: A, an $m \times n$ matrix of 0's and 1's
Find: $\vec{x}$, a n-vector of 0's and 1's such that $A\vec{x} = \vec{1}$.



$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

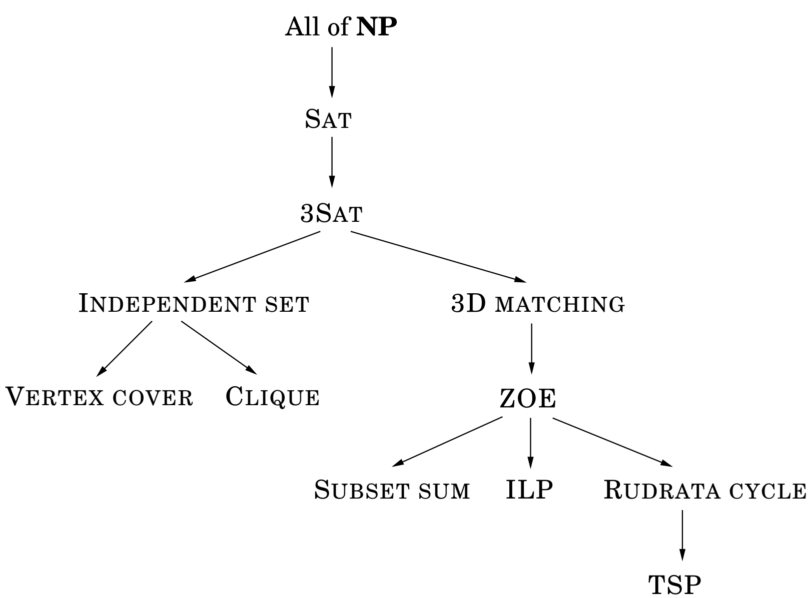
Construction: Example

$R = \{ t_1, \ldots, t_5 \}$ where
t_1 = (Al,Alice,Armadillo)
t_2 = (Chet,Beatrice,Canary)
t_3 = (Chet,Carol,Bobcat)
t_4 = (Bob,Carol,Bobcat)
t_5 = (Bob,Alice,Armadillo)

	t_1	t_2	t_3	t_4	t_5
Al	1	0	0	0	0
Bob	0	0	0	1	1
Chet	0	1	1	0	0
Alice	1	0	0	0	1
Beatrice	0	1	0	0	0
Carol	0	0	1	1	0
Armadilo	1	0	0	0	1
Bobcat	0	0	1	1	0
Canary	0	1	0	0	0

$A \times (1,1,1,0,0) = (1,0,2,1,1,1,1,1,1)$
$A \times (1,1,0,1,0) = (1,1,1,1,1,1,1,1,1)$

The Plan of §8.3



ZOE $\trianglelefteq$ Subset Sum

Zero-One Equations (ZOE)

Given: A, an $m \times n$ matrix of 0's and 1's
Find: $\vec{x}$, a n-vec. of 0's and 1's $\ni A\vec{x} = \vec{1}$.

Subset Sum

Given: A multiset M and goal G, all ints.
Find: An $M' \subseteq M$ such that $G = \sum_{x \in M'} x$.

A Sample Reduction

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad \trianglelefteq \quad \begin{cases} M = \{ 1000_6, 0001_6, 0110_6, 1000_6, 0100_6 \} \\ G = 1111_6 \end{cases}$$

$1111_6 = 1 * 6^3 + 1 * 6^2 + 1 * 6^1 + 1 * 6^0$, etc.

ZOE $\trianglelefteq$ ILP

Zero-One Equations (ZOE)

Given: A, an $m \times n$ matrix of 0's and 1's
Find: $\vec{x}$, a n-vec. of 0's and 1's $\ni A\vec{x} = \vec{1}$.

Int. Linear Programming (ILP)

Given: constraints $A\vec{x} \leq \vec{b}$
Find: A vector of integers $\vec{x} \ni A\vec{x} \leq \vec{b}$.

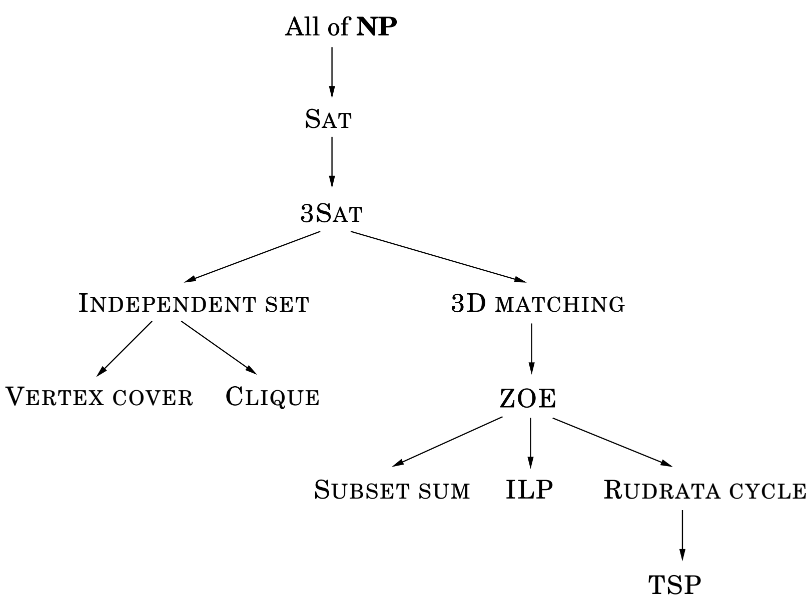
Reduction:

- Each ZOE equation: $\vec{a} \cdot \vec{x} = 1$
  is rewritten to two inequalities: $\vec{a} \cdot \vec{x} \leq 1$ and $-\vec{a} \cdot \vec{x} \leq -1$
- Each ZOE variable x_i gets two additional constraints: $x_i \leq 1$ and $-x_i \leq 0$.

ZOE $\trianglelefteq$ Rudrata Cycle

See the text for this one.
It is epic proof.

The Plan of §8.3



Rudrata/Hamiltonian Cycle $\trianglelefteq$ TSP

Rudrata/Hamiltonian Cycle Problem

Given: $G = (V, E)$, an undirected graph.
Find: A simple cycle that visits each vertex of G.

Traveling Salesman Problem (TSP)

Given: V', n vertices; all $\frac{n\cdot(n-1)}{2}$-many distances between them; and b, a budget
Find: π, an ordering of $1, \ldots, n$, such that $\sum_{i=1}^{n} d_{\pi(i),\pi(1+(i \bmod n))} \leq b$

Construction Pick $\alpha \geq 1$.
Given (V, E), define

$$V' = V$$

$$d_{i,j} = \begin{cases} 1, & \text{if } (i,j) \in E; \\ 1+\alpha, & \text{otherwise.} \end{cases}$$

$$b = |V|$$

Claim: (V, E) has a R/H cycle
$\iff$ (V', d) has tour of cost $\leq b$.

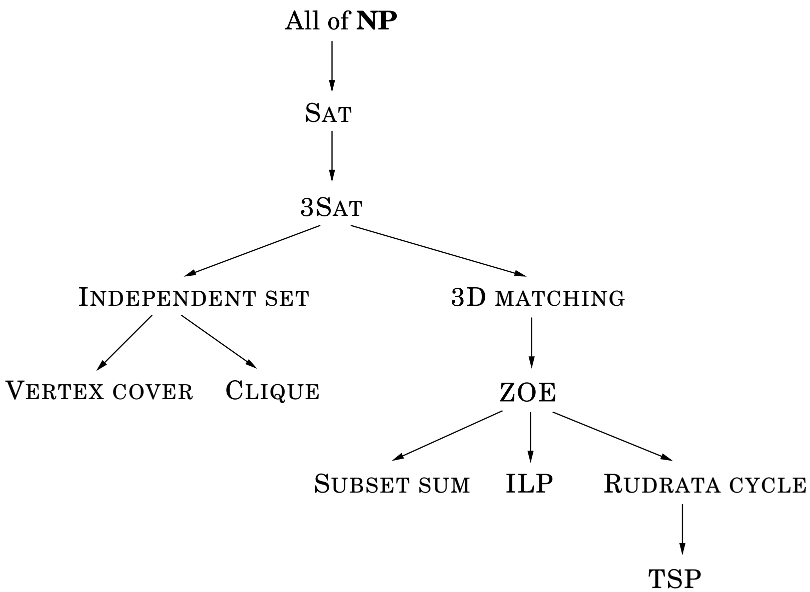
If $\alpha = 1$**:**

- Distances satisfy the triangle inequality: $d_{ij} + d_{jk} \geq d_{ik}$.
- These instance of TSP are approximatable (Chap. 9).

If $\alpha \gg 1$**:**

- Gap: either a solution of cost b, or solutions with costs $\geq b + \alpha$, but none inbetween.

The Plan of §8.3



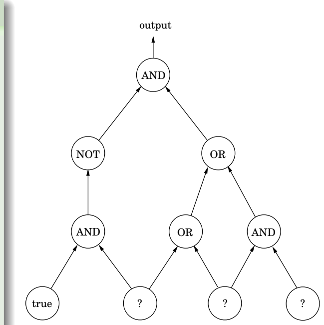
All of NP $\trianglelefteq$ Circuit SAT, 1

Circuit SAT

Given: A Boolean circuit, a dag with five types of gates

1. AND *gates* and OR *gates* (indegree 2)
2. NOT *gates* (indegree 1)
3. *Known input gates* (indegree 0) labeled True or False.
4. *Unknown inputs gates* (indgree 0) labeled "?".
5. An *output gate* (a particular sink)

Find: A truth assignment to the unknown input gates so that the circuit evaluates to True.



Claim 1: SAT $\trianglelefteq$ Circuit SAT

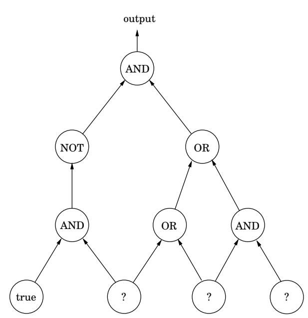
All of NP $\trianglelefteq$ Circuit SAT, 2

Circuit SAT

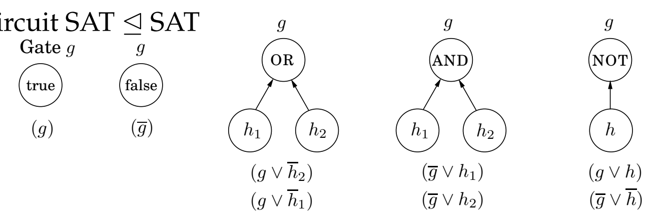
Given: A Boolean circuit, a dag with five types of gates

1. AND *gates* and OR *gates* (indegree 2)
2. NOT *gates* (indegree 1)
3. *Known input gates* (indegree 0) labeled True or False.
4. *Unknown inputs gates* (indgree 0) labeled "?".
5. An *output gate* (a particular sink)

Find: A truth assignment to the unknown input gates so that the circuit evaluates to True.



Claim 2: Circuit SAT $\trianglelefteq$ SAT



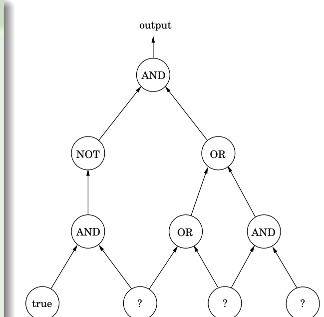
All of NP $\trianglelefteq$ Circuit SAT, 3

Circuit SAT

Given: A Boolean circuit, a dag with five types of gates

1. AND *gates* and OR *gates* (indegree 2)
2. NOT *gates* (indegree 1)
3. *Known input gates* (indegree 0) labeled True or False.
4. *Unknown inputs gates* (indgree 0) labeled "?".
5. An *output gate* (a particular sink)

Find: A truth assignment to the unknown input gates so that the circuit evaluates to True.



Recall from Chapter 7: Any polytime algorithm can be expressed as a boolean circuit (scaled to the input size). Dirty trick warning!!!

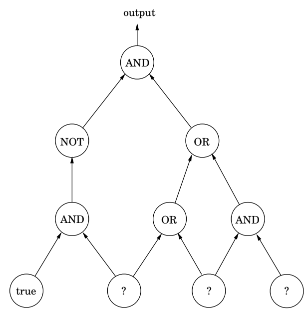
All of NP ⊴ Circuit SAT, 4

Circuit SAT

Given: A Boolean circuit, a dag with five types of gates

1. AND *gates* and OR *gates* (indegree 2)
2. NOT *gates* (indegree 1)
3. *Known input gates* (indegree 0) labeled True or False.
4. *Unknown inputs gates* (indgree 0) labeled “?”.
5. An *output gate* (a particular sink)

Find: A truth assignment to the unknown input gates so that the circuit evaluates to True.



Given an NP-problem Q with polytime checking function C.

- For instance I, build the boolean circuit B that checks I, with “?” labeling the “solution input gates.”
- Then I has a Q-solution $\iff$ B has a satisfying assignment.

$\therefore$ ...

All of NP ⊴ Circuit SAT, 5

Theorem (The Cook-Levin Theorem)

SAT is NP-complete.

Next: Dealing with NP-completeness