BU CS 332 – Theory of Computation

Lecture 24:

- More NP-completeness
- Space complexity (?)

Reading:

Sipser Ch 7.4-7.5,
8.1-8.2

Mark Bun

April 26, 2021

Polynomial-time reducibility

Definition:

A function $f: \Sigma^* \to \Sigma^*$ is polynomial-time computable if there is a polynomial-time TM M which, given as input any $w \in \Sigma^*$, halts with only $f(w)$ on its tape.

Definition:

Language A is polynomial-time reducible to language B, written

$$A \leq_{\mathrm{p}} B$$

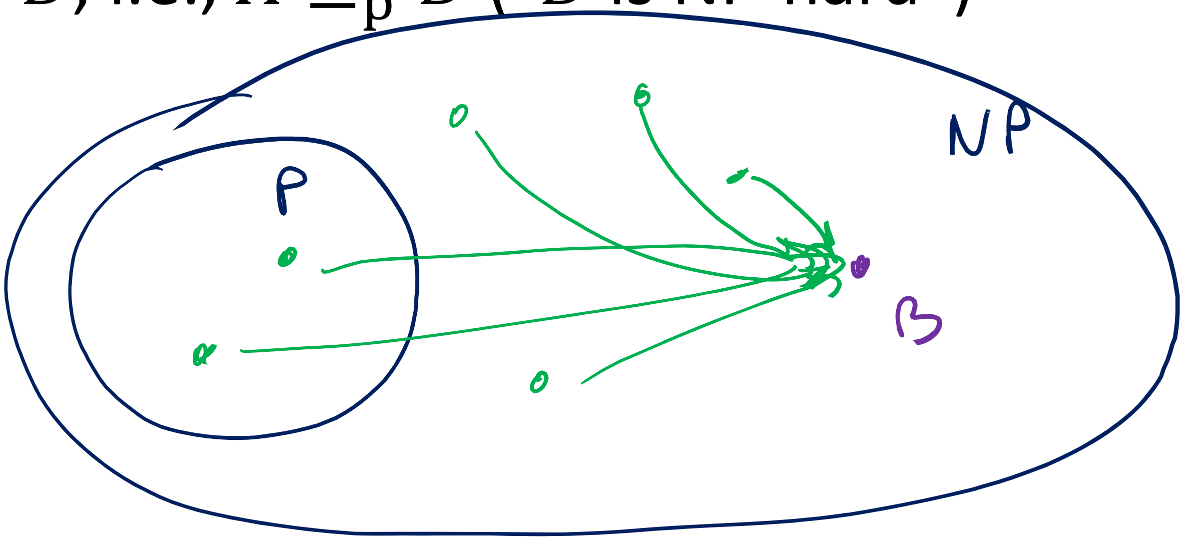
if there is a polynomial-time computable function $f: \Sigma^* \to \Sigma^*$ such that for all strings $w \in \Sigma^*$, we have $w \in A \Leftrightarrow f(w) \in B$

NP-completeness

“The hardest languages in NP”

Definition: A language B is NP-complete if

1) $B \in \mathrm{NP}$, and
2) Every language $A \in \mathrm{NP}$ is poly-time reducible to B, i.e., $A \leq_{\mathrm{p}} B$ (“B is NP-hard”)



The usual way to prove NP-completeness

Theorem:

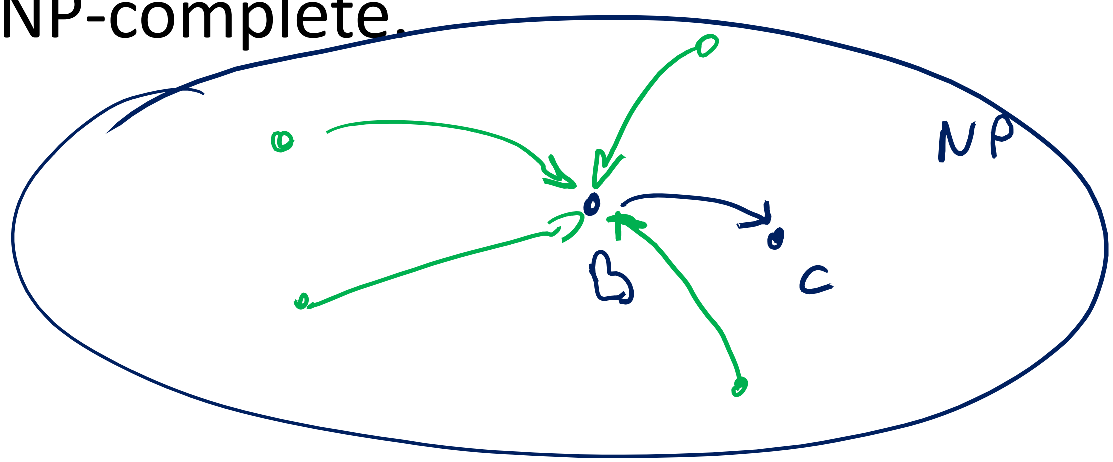
If

1) $C \in \mathrm{NP}$, and

2) There is an NP-complete language B such that

$B \leq_{\mathrm{p}} C$

then C is NP-complete.



Some general reduction strategies

- Reduction by simple equivalence

  Ex. $IND-SET \leq_{\mathrm{p}} VERTEX-COVER$

  $VERTEX-COVER \leq_{\mathrm{p}} IND-SET$

- Reduction from special case to general case

  Ex. $VERTEX-COVER \leq_{\mathrm{p}} SET-COVER$

  $3SAT \leq_{\mathrm{p}} SAT$

- "Gadget" reductions

  Ex. $SAT \leq_{\mathrm{p}} 3SAT$

  $\boxed{3SAT \leq_{\mathrm{p}} IND-SET}$

$3SAT$ (3-CNF Satisfiability)

Definitions:

- A literal either a variable of its negation x_5 , $\overline{x_7}$
- A clause is a disjunction (OR) of literals Ex. $x_5 \vee \overline{x_7} \vee x_2$
- A 3-CNF is a conjunction (AND) of clauses where each clause contains exactly 3 literals

Ex. $C_1 \wedge C_2 \wedge \ldots \wedge C_m =$

$$(x_5 \vee \overline{x_7} \vee x_2) \wedge (\overline{x_3} \vee x_4 \vee x_1) \wedge \cdots \wedge (x_1 \vee x_1 \vee x_1)$$

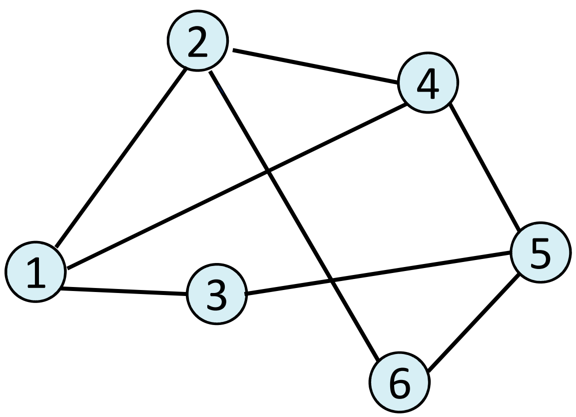
$3SAT = \{\langle\varphi\rangle | \varphi \text{ is a satisfiable } 3-\text{CNF}\}$

Last time: $3SAT$ is NP-complete

Independent Set

An **independent set** in an undirected graph G is a set of vertices that includes at most one endpoint of every edge.

$$IND-SET = \{\langle G, k\rangle | G \text{ is an undirected graph containing an independent set with} \geq k \text{ vertices}\}$$



Which of the following are independent sets in this graph?

a) $\{1\}$
b) $\{1, 5\}$
~~c)~~ $\{2, 3, 6\}$
d) $\{3, 4, 6\}$

Independent Set is NP-complete

1) $IND-SET \in \mathrm{NP}$
2) Reduce $3SAT \leq_{\mathrm{p}} IND-SET$

Proof of 1) The following gives a poly-time verifier for $IND-SET$

Certificate: Vertices $v_1, \ldots, v_k$

Verifier:

"On input $\langle G, k; v_1, \ldots, v_k \rangle$, where G is a graph, k is a natural number,

1. Check that $v_1, \ldots v_k$ are distinct vertices in G
2. Check that there are no edges between the v_i's."

Independent Set is NP-complete

1) $IND - SET \in \text{NP}$
2) Reduce $3SAT \leq_{\text{p}} IND - SET$

instance of 3SAT

instance of IND-SET

$f : \langle \varphi \rangle \mapsto \langle G, k \rangle$

$\langle \varphi \rangle \in 3SAT \Rightarrow G$ has an ind. set of size $\geq k$

$\langle \varphi \rangle \notin 3SAT \Rightarrow G$ has no ind. set of size k

Proof of 2) The following TM computes a poly-time reduction.

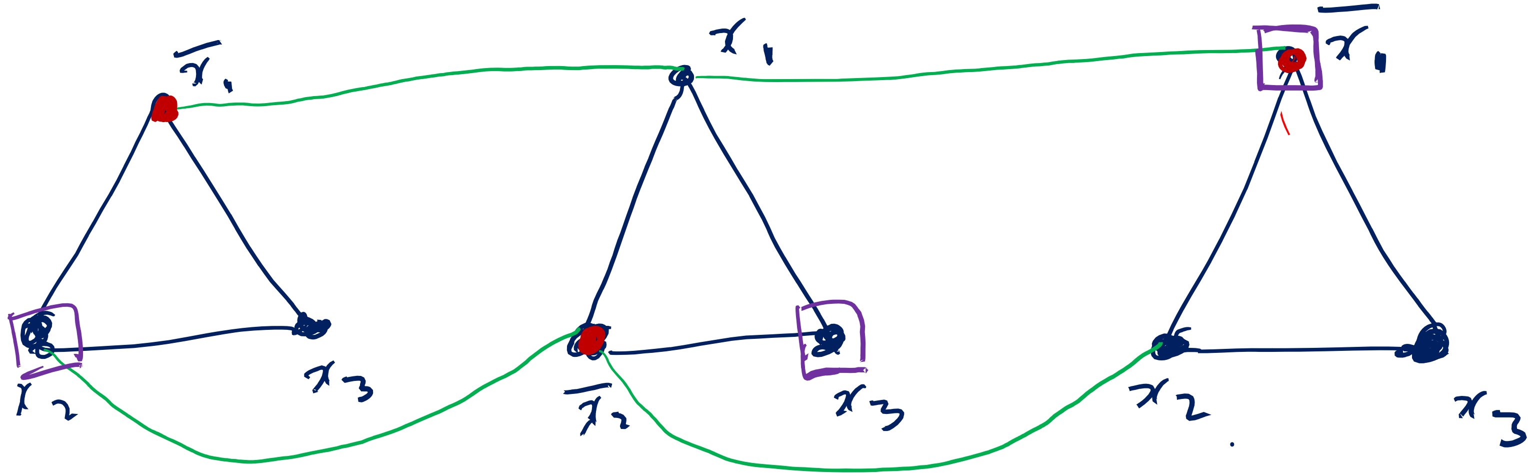
"On input $\langle \varphi \rangle$, where φ is a 3CNF formula,

1. Construct graph G from φ
   - G contains 3 vertices for each clause, one for each literal.
   - Connect 3 literals in a clause in a triangle.
   - Connect every literal to each of its negations.
2. Output $\langle G, k \rangle$, where k is the number of clauses in φ."

Example of the reduction

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

$G =$



$k = 3$

<u>Output</u>: $\langle G, k \rangle$

Ex: $\Rightarrow$ $x_1 = 0$, $x_2 = 0$, $x_3 = 1$ is a sat. ass'm't $\Rightarrow$ ind set of size 3

$\Leftarrow$ S = D's

$x_2 = 1$
$x_3 = 1$
$x_1 = 0$
is a sat. ass'm't

Proof of correctness for reduction

Let k = # clauses and l = # literals in φ

Correctness: φ is satisfiable iff G has an independent set of size k

$\Longrightarrow$ Given a satisfying assignment, select one true literal from each triangle. This is an independent set of size k

$\Longleftarrow$ Let S be an independent set in G of size k

- S must contain exactly one vertex in each triangle
- Set these literals to true, and set all other variables arbitrarily
- Truth assignment is consistent and all clauses are satisfied

Runtime: $O(k + l^2)$ which is polynomial in input size

Some general reduction strategies

- Reduction by simple equivalence

  Ex. $$\boxed{\begin{array}{l} IND-SET \le_{\mathrm{p}} VERTEX-COVER \\ VERTEX-COVER \le_{\mathrm{p}} IND-SET \end{array}}$$

- Reduction from special case to general case

  Ex. $VERTEX-COVER \le_{\mathrm{p}} SET-COVER$

  $3SAT \le_{\mathrm{p}} SAT$

- “Gadget” reductions

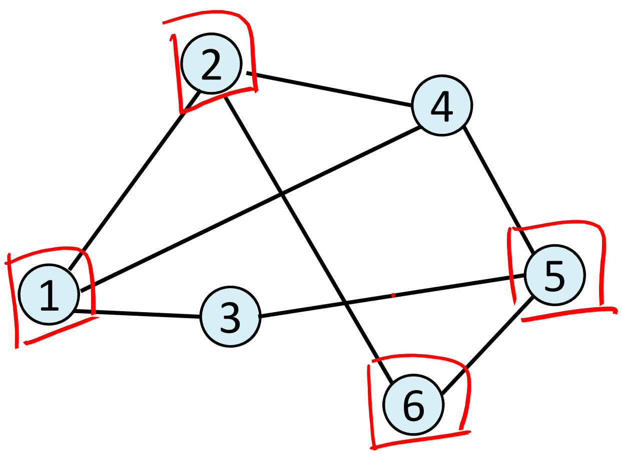
  Ex. $SAT \le_{\mathrm{p}} 3SAT$

  $3SAT \le_{\mathrm{p}} IND-SET$

Vertex Cover

Given an undirected graph G, a **vertex cover** in G is a subset of nodes which includes at ***least*** one endpoint of every edge.

$VERTEX - COVER$ = $\{\langle G, k\rangle$ | G is an undirected graph which has a vertex cover with $\leq k$ vertices$\}$



Which of the following are vertex covers in this graph?

~~a)~~ $\{1\}$
~~b)~~ $\{1, 6\}$
c) $\{1, 2, 5\}$
d) $\{1, 2, 5, 6\}$

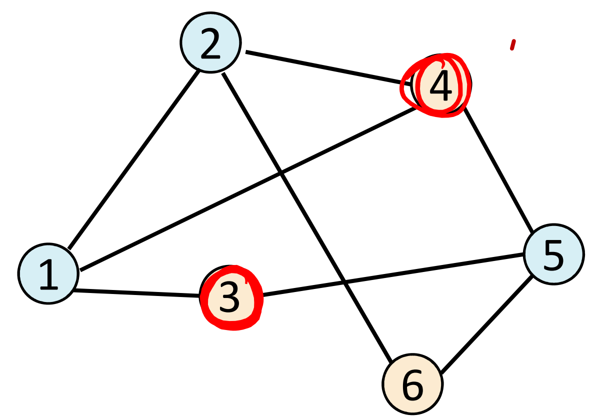
Independent Set and Vertex Cover

Claim. S is an independent set iff $V \setminus S$ is a vertex cover.

Corollary: G has an ind. set of size $\geq k$ $\iff$ G has a vertex cover of size $\leq |V| - k$

$\Longrightarrow$ Let S be any independent set. S

- Consider an arbitrary edge (u, v).
- S is independent $\Longrightarrow u \notin S$ or $v \notin S$ $\Longrightarrow$ $u \in V \setminus S$ or $v \in V \setminus S$.
- Thus, $V \setminus S$ covers (u, v).



$\Longleftarrow$ Let $V \setminus S$ be any vertex cover. $V \setminus S$

- Consider two nodes $u \in S$ and $v \in S$.
- Then $(u, v) \notin E$ since $V \setminus S$ is a vertex cover.
- Thus, no two nodes in S are joined by an edge $\Longrightarrow S$ is an independent set.

INDEPENDENT SET reduces to VERTEX COVER

$A \le_p B$: $w \in A \iff f(w) \in B$

Theorem. IND-SET $\leq_p$ VERTEX-COVER.

What do we need to do to prove this theorem?

a) Construct a poly-time nondet. TM deciding IND-SET
b) Construct a poly-time deterministic TM deciding IND-SET
c) Construct a poly-time nondet. TM mapping instances of IND-SET to instances of VERTEX-COVER
d) Construct a poly-time deterministic TM mapping instances of IND-SET to instances of VERTEX-COVER
e) Construct a poly-time nondet. TM mapping instances of VERTEX-COVER to instances of IND-SET
f) Construct a poly-time deterministic TM mapping instances of VERTEX-COVER to instances of IND-SET

INDEPENDENT SET reduces to VERTEX COVER

Theorem. IND-SET $\leq_{\mathrm{p}}$ VERTEX-COVER.

Proof. The following TM computes the reduction.

"On input $\underbrace{\langle G, k \rangle}_{\text{IND-SET instance}}$, where G is an undirected graph and k is an integer,

1. Output $\underbrace{\langle G, n - k \rangle}_{\text{VERTEX-COVER instance}}$, where n is the number of nodes in G."

Correctness:

- G has an independent set of size at least k iff it has a vertex cover of size at most $n - k$.

Runtime:

- Reduction runs in linear time.

VERTEX COVER reduces to INDEPENDENT SET

Theorem. VERTEX-COVER $\leq_{\mathrm{p}}$ IND-SET

Proof. The following TM computes the reduction.

"On input $\underbrace{\langle G, k\rangle}_{\text{instance of VERTEX-COVER}}$, where G is an undirected graph and k is an integer,

1. Output $\underbrace{\langle G, n-k\rangle}_{\text{instance of IND-SET}}$, where n is the number of nodes in G."

Correctness:

- G has a vertex cover of size at most k iff it has an independent set of size at least $n-k$.

Runtime:

- Reduction runs in linear time.

A Brief Tour of Space (Complexity)

Space analysis

Space complexity of a TM (algorithm) = maximum number of tape cells it uses on a worst-case input

Formally: Let $f : \mathbb{N} \to \mathbb{N}$. A TM M runs in space $f(n)$ if on every input $w \in \Sigma^{\cancel{*}n}$, M halts on w using at most $f(n)$ cells

For nondeterministic machines: Let $f : \mathbb{N} \to \mathbb{N}$. An NTM N runs in space $f(n)$ if on every input $w \in \Sigma^{\cancel{*}n}$, N halts on w using at most $f(n)$ cells on every computational branch

Space complexity classes

Let $f : \mathbb{N} \to \mathbb{N}$

A language $A \in \text{SPACE}(f(n))$ if there exists a basic single-tape (deterministic) TM M that

1) Decides A, and
2) Runs in space $O(f(n))$

A language $A \in \text{NSPACE}(f(n))$ if there exists a single-tape nondeterministic TM N that

1) Decides A, and
2) Runs in space $O(f(n))$

Space vs. Time

$$\mathrm{TIME}(f(n)) \subseteq \mathrm{NTIME}(f(n))$$
$$\subseteq \mathrm{SPACE}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$$

How about the opposite direction? Can low-space algorithms be simulated by low-time algorithms?

Theorem: A TM running in space $f(n)$ also runs in time $2^{O(f(n))}$

TM uses space f(n) $\Rightarrow$ TM can be in at most $2^{O(f(n))}$ configurations

$\Rightarrow$ TM cannot run for more than $2^{O(f(n))}$ steps w/o looping

$$\boxed{\mathrm{SPACE}(f(n)) \subseteq \mathrm{TIME}(2^{O(f(n))})}$$

Savitch's Theorem: Deterministic vs. Nondeterministic Space

1970

Theorem: Let f be a function with $f(n) \geq \log n$. Then $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}\left((f(n))^2\right)$.

$\mathrm{NSPACE}(n^2) \subseteq \mathrm{SPACE}(n^4)$

Complexity class PSPACE

Definition: PSPACE is the class of languages decidable in polynomial space on a basic single-tape (deterministic) TM

$$\mathrm{PSPACE} = \bigcup_{k=1}^{\infty} \mathrm{SPACE}(n^k)$$

Definition: $\mathrm{NPSPACE}$ is the class of languages decidable in polynomial space on a single-tape (nondeterministic) TM

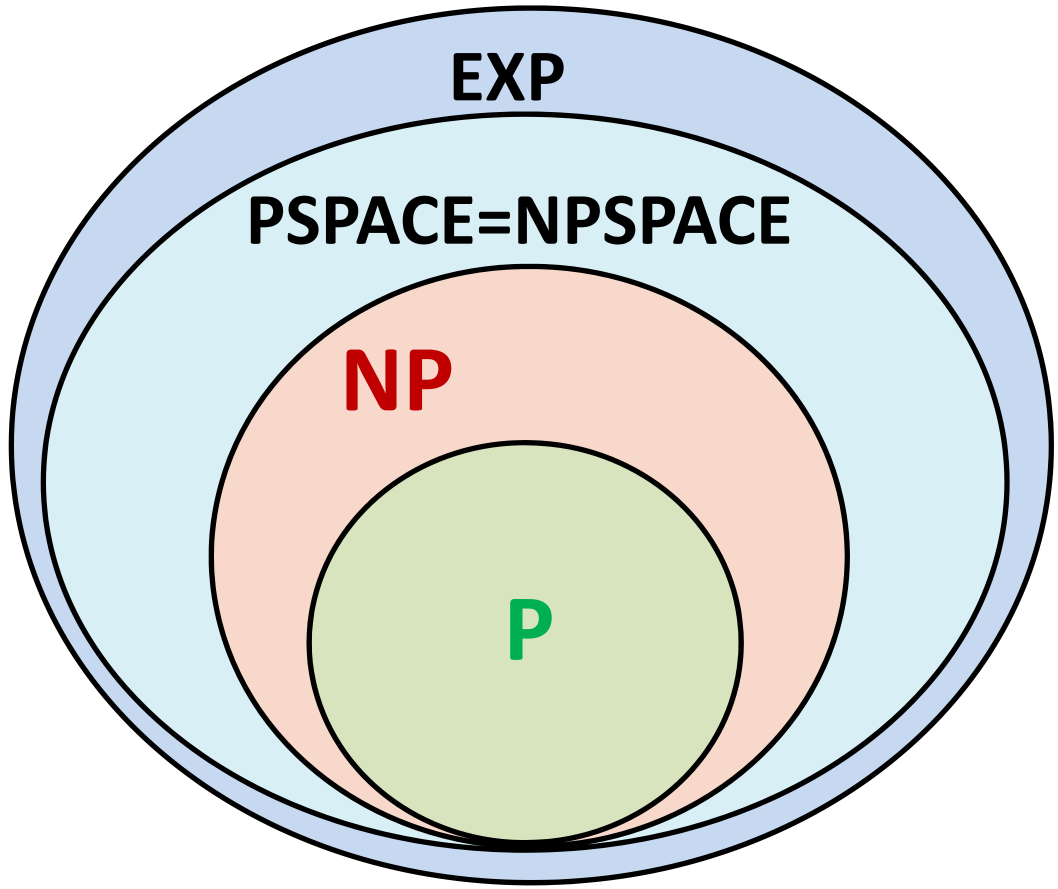
$$\mathrm{NPSPACE} = \bigcup_{k=1}^{\infty} \mathrm{NSPACE}(n^k)$$

PSPACE = NPSPACE (via savitch)

Relationships between complexity classes

1. $\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}$

since $\mathrm{SPACE}\big(f(n)\big) \subseteq \mathrm{TIME}(2^{O(f(n))})$



2. $\mathrm{P} \neq \mathrm{EXP}$

   (via time hierarchy)

Which containments in (1) are proper?

Unknown!

Course Evaluations

bu.campuslabs.com/courseeval