

Questions

- How good is computer security?
- What would computer security professionals and hackers pursue?

CSC460 D2

1

Unit D2: Overview

- Notice the connection among NP problems
- Understand polynomial time reducibility
- Identify and analyze a special class of NP problems
- Preview Exercise D2 "NP/NPC Practice; Mini Research Ideas"

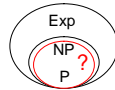
CSC460 D2

2

NP Problems

Review

- Class of Nondeterministic Polynomial-time problems
 - E.g., SAT, TSP, and many others
- Definition #1: A solution can be **verified in polynomial time**.
- Definition #2: A **nondeterministic TM** can decide in **polynomial time**.



CSC460 D2

3

Sample NP Problems

Review

- Digital Circuits (#13) = **SATISFIABILITY (SAT)**
- Cross-country interviews (#16) = **TRAVELING SALESPERSON (TSP)**
- Cryptography (#11)
- Monkey Puzzle (#12)
- Professor Assignment (#14)
- Knapsack Problem (#15)
- CPU Register Allocation (#17)
- Map Coloring (#18)

CSC460 D2

4

Standard NP Problems

- SATISFIABILITY (SAT)
- 3SAT
- TRAVELING SALEPERSON (TSP)
- HAMILTONIAN CIRCUIT (HC)
- VERTEX COVER (VC)
- CLIQUE
- 3D MATCHING (3DM)
- PARTITION

CSC460 D2

5

SATISFIABILITY (SAT)

- Informally, finding a satisfying truth-value assignment for a conjunction of (disjunctive) clauses, e.g., $(p \text{ or } q) \text{ and } (p \text{ or } (\text{not } q)) \text{ and } ((\text{not } p) \text{ or } r \text{ or } s)$.
 - Note: A clause is a disjunction of propositional (true/false) variables or its negation.
- Formally, $\{(C, V, a) \mid \text{set of (disjunctive) clauses } C \text{ with a finite set of variables } V \text{ such that assignment } a: V \rightarrow \{\mathbf{T}, \mathbf{F}\} \text{ satisfies all the clauses in } C\}$

CSC460 D2

6

3SAT

- A special case of SAT with the pattern (or or) and (or or) and ... and (or or) where each ' ' holds a variable (e.g., p) or the negation of a variable (e.g., **not** q)
 - I.e., each (...) is **limited to 3 things**
- Connection to SAT?

CSC460 D2

7

TRAVELING SALESPERSON (TSP)

- Informally, finding a tour of all the cities (without repeating) within the set budget.
 - Formally, $\{(\mathbf{G}, c, b) \mid \text{graph } \mathbf{G} = (V, E) \text{ with the cost function } c \text{ such that a tour is possible within the total cost of } b\}$
 - **tour**: $(v_1, \dots, v_{k=|V|})$ such that
 - $v_i \in V$ for every $1 \leq i \leq k$,
 - $v_i \neq v_j$ for every pair $i \neq j$,
 - $(v_i, v_{i+1}) \in E$ for every $1 \leq i < k$,
 - $(v_k, v_1) \in E$
- completeness of G?**

CSC460 D2

8

HAMILTONIAN CIRCUIT (HC)

- Informally, finding a closed circuit to travel all the points without repeating.
- Formally, $\{\mathbf{G} \mid \text{graph } \mathbf{G} = (V, E) \text{ admits a hamiltonian circuit}\}$
- **hamiltonian circuit**: $(v_1, \dots, v_{k=|V|})$ such that
 - $v_i \in V$ for every $1 \leq i \leq k$,
 - $v_i \neq v_j$ for every pair $i \neq j$,
 - $(v_i, v_{i+1}) \in E$ for every $1 \leq i < k$,
 - $(v_k, v_1) \in E$
- Connection to TSP? Hamiltonian Path - between two points
Euler circuit - traversing all the edges

CSC460 D2

9

VERTEX COVER (VC)

- Informally, finding a graph segment of size k or less that contain at least one point of every edge.
- Formally, $\{(\mathbf{G}, k) \mid \text{graph } \mathbf{G} = (V, E) \text{ admits a vertex cover where } 0 < k \leq |V|\}$
- **vertex cover**: $W \subseteq V$ such that
 - $|W| \leq k$,
 - For every $(u, v) \in E$, $u \in W$ or $v \in W$

CSC460 D2

10

CLIQUE

- Informally, finding a subgraph of size k or less such that every two nodes are connected by an edge in the graph.
- Formally, $\{(\mathbf{G}, k) \mid \text{graph } \mathbf{G} = (V, E) \text{ contains a } k\text{-clique where } 0 < k \leq |V|\}$
- **k-clique**: $W \subseteq V$ such that
 - $|W| = k$,
 - For every $u, v \in W$, $(u, v) \in E$
- Connection to VC?

CSC460 D2

11

Marriage Problem

Intermission

- Informally, finding a complete list of male-female pairs (without polygamy)
- Formally, $\{(R, k) \mid R \subseteq M \times F \text{ contains a matching where } M, F \text{ are disjoint and } |M| = |F| = k\}$
- **matching**: $P \subseteq R$ such that
 - $|P| = k$,
 - $\forall (x, y), (u, v) \in P [(x \neq u) \wedge (y \neq v)]$
 - (No two elements of P agree in any coordinate)

CSC460 D2

NP?

12

3D MATCHING (3DM)

- Informally, a variant of the marriage problem involving 3 different sexes.
- Formally, $\{(R, k) \mid R \subseteq W \times X \times Y \text{ contains a matching where } W, X, Y \text{ are disjoint and } |W| = |X| = |Y| = k\}$
- matching: $M \subseteq R$ such that
 - $|M| = k$,
 - $\forall (x, y, z), (u, v, w) \in M [(x \neq u) \wedge (y \neq v) \wedge (z \neq w)]$
 - (No two elements of M agree in any coordinate)

CSC460 D2

13

PARTITION

- Informally, finding a subset that divides a set evenly with respect to the members' "prices."
- Formally, $\{(A, s) \mid \text{finite set } A \text{ with a balanced partition}\}$
- balanced partition: $B \subseteq A$ such that
 - $p: A \rightarrow \mathbf{Z}^+$ ("price" function)

$$\sum_{a \in B} p(a) = \sum_{a \in (A-B)} p(a)$$

CSC460 D2

connection among problems? 14

Connection among Problems

- Special cases
 - 3SAT (clauses of size 3) is a special case of SAT
 - HC is a special case of TSP (with weight)
- Similarity
 - VC and CLIQUE
- Can some problems be reduced to others?

Requirements for reduction?

CSC460 D2

15

Polynomial Time Reducibility

- A is polynomial time reducible (P -reducible) to B .
 - A is reducible to B .
 - This can be done in polynomial time.
- Transfer of properties
 - B has a positive property $\Rightarrow A$ has it too.
 - Example: If $B \in P, A \in P$.
 - A has a negative property $\Rightarrow B$ has it too.
 - Example: If $A \notin P, B \notin P$.

CSC460 D2

16

NP-Complete (NPC) Problems

- Definition
 - The problem is in NP.
 - Any NP problem can be P -reduced to that problem. [i.e., can solve any NP problem]



- Generalization of X -complete
 - P -complete, Turing-complete, AI-complete

CSC460 D2

17

Cook-Levin Theorem

- SAT is NP-complete.
 - SAT is in NP.
 - Any NP problem can be P -reduced to SAT.

Challenge

- Generic NP problem representation

Terminology

$(p \text{ or } q)$ and $(p \text{ or } (\text{not } q))$ and $((\text{not } p) \text{ or } r)$
 clause clause clause

CSC460 D2

18

Proof Outline

- Lemma: Any NP problem can be P-reduced to SAT.
- Components
 - An arbitrary polynomial time Nondeterministic TM (NTM) program (or **verification sequence of an answer**) can be reduced to SAT.
 - Acceptable strings \rightarrow True SAT statements
 - Acceptable strings \leftarrow True SAT statements
 - The reduction is in P (polynomial reduction).
 - Complexity analysis

Garey & Johnson 1979
(standard NPC reference)

Polynomial Time Conditions

- Cannot spend exponential amount of time (cf. the input)
- Cannot use an exponential amount of tape space
 - The limit to polynomial tape space does not guarantee P. But exponential space cannot be handled in P.
- Cannot use an exponential number of symbols

Reduction Idea

Introduce *propositional variables* as follows:

- Simulating **states**
 - TM is in state q_k at time $i \Rightarrow Q_{(i,k)}$
- Simulating **head** movements
 - Head is at tape position j at time $i \Rightarrow H_{(i,j)}$
- Simulating **tape symbols**
 - Tape symbol of position j is s_k at time $i \Rightarrow S_{(i,j,k)}$

Polynomially bounded?

Imposing Logical Conditions

- At time 0, TM is in its initial configuration.
- At time i , TM
 - Is in exactly one state, **view as verification with a NTM**
 - Is at exactly one tape position, and
 - Reads exactly one tape symbol, and
 - Moves to the next configuration defined by δ .
- Within a polynomially-bounded time, TM enters the final configuration.

- States $\Rightarrow Q_{(i,k)}$
- Head movements $\Rightarrow H_{(i,j)}$
- Tape symbols $\Rightarrow S_{(i,j,k)}$

3SAT Statements

- States $\Rightarrow Q_{(i,k)}$
- Head movements $\Rightarrow H_{(i,j)}$
- Tape symbols $\Rightarrow S_{(i,j,k)}$
- \wedge = 'and', \vee = 'or', \neg = 'not'

- At time 0, TM is in its initial configuration. $\Rightarrow Q_{(0,0)} \wedge H_{(0,0)} \wedge S_{(0,0,q_0)} \wedge S_{(0,1,q_1)} \wedge \dots \wedge S_{(0,n-1,q_{n-1})} \wedge S_{(0,n,0)} \wedge \dots \wedge S_{(0,\text{poly-bound},0)}$
- At time i , TM
 - Is in exactly one state \Rightarrow at least two & at most one $\Rightarrow (Q_{(i,0)} \vee \dots \vee Q_{(i,\text{poly-bound})}) \wedge \neg(Q_{(i,j)} \wedge Q_{(i,j')}) \wedge \dots$ [for all $j \neq j'$] **all clauses?**
 - Moves to a next configuration defined by δ .
 - Case 1 (at j at time i):

$$((H_{(i,j)} \wedge Q_{(i,k)} \wedge S_{(i,j,k)}) \rightarrow H_{(i+1,j+\delta)} \wedge (\dots \vee Q_{(i+1,k')}) \wedge (\dots \vee S_{(i+1,j',r')})$$
 Note $((H_{(i,j)} \wedge Q_{(i,k)} \wedge S_{(i,j,k)}) \rightarrow H_{(i+1,j+\delta)} \wedge (\dots \vee Q_{(i+1,k')}) \wedge (\dots \vee S_{(i+1,j',r')}) \Leftrightarrow (\neg H_{(i,j)} \vee \neg Q_{(i,k)} \vee \neg S_{(i,j,k)} \vee H_{(i+1,j+\delta)} \vee H_{(i+1,j',r')})$
 - Case 2 (not at j at time i): i.e., no change

$$((\neg H_{(i,j)} \wedge S_{(i,j,k)}) \rightarrow S_{(i+1,j,k)})$$
 Note $((\neg H_{(i,j)} \wedge S_{(i,j,k)}) \rightarrow S_{(i+1,j,k)}) \Leftrightarrow (H_{(i,j)} \vee \neg S_{(i,j,k)} \vee S_{(i+1,j,k)})$

Cook-Levin Theorem Summary

SAT is NP-complete.

1. SAT is in NP. \Leftarrow Truth value evaluation
2. Any NP problem can be P-reduced to SAT.
 - An arbitrary NTM program in P is polynomial time reducible to SAT.
 - Simulate states, head movements, and tape symbols as a logical condition
 - Construct the statement in polynomial time

3SAT is NPC

- 3SAT is in NP.
- SAT is polynomially reducible to 3SAT.
 - If 3SAT can solve SAT, 3SAT is NPC.
- Reduction idea
 - Transform each clause into an equivalent collection of clauses with 3 variables
 - Computability: Existence of an algorithm
 - Tractability: Polynomial bound

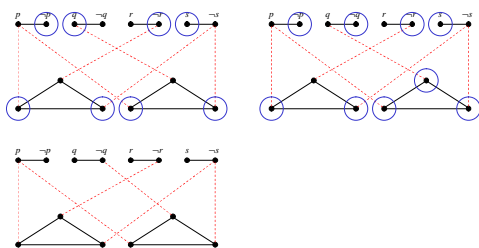
How about 2SAT?

VC is NPC

- 3SAT is polynomially reducible to VC.
- Construct a graph with some k that would translate 3SAT to VC
- Example
 - $(p \text{ or } (\text{not } r) \text{ or } (\text{not } s)) \text{ and } (p \text{ or } q \text{ or } (\text{not } s))$
 - $(p \vee \neg r \vee \neg s) \wedge (p \vee q \vee \neg s)$
- Polynomial reduction to CLIQUE possible

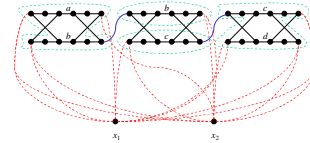
OK to use this special VC?

3SAT to VC

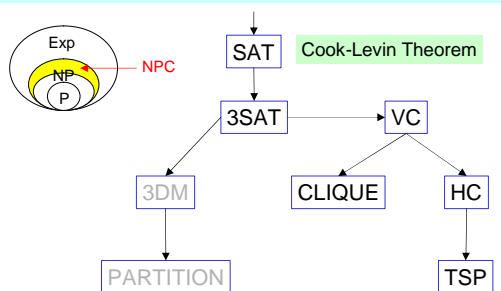


HC is NPC

- VC is polynomially reducible to HC.
- ... a bit complicated...

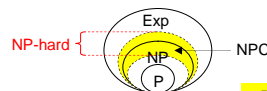


NPC Summary



NP-Hard Problems

- Definition
 - Any NP problem can be reduced to the problem. [at least as difficult as NPC]
 - Problem is *not necessarily* in NP. [OK to be in NP \Rightarrow NP-hard includes NPC]



X-Hard for any class X

- Define NPC with NP-hard
- Type of acceptable reductions?

Unit Summary

- A whole bunch of practical problems are in NPC.
- All NPC problems are equivalent with respect to their time complexity via polynomial reduction.
- A polynomial solution to any of these NPC problem would conclude $P = NP$.

Summary Question

- Do you understand the significance of NPC problems? Explain.
- Questions/Comments/Suggestions