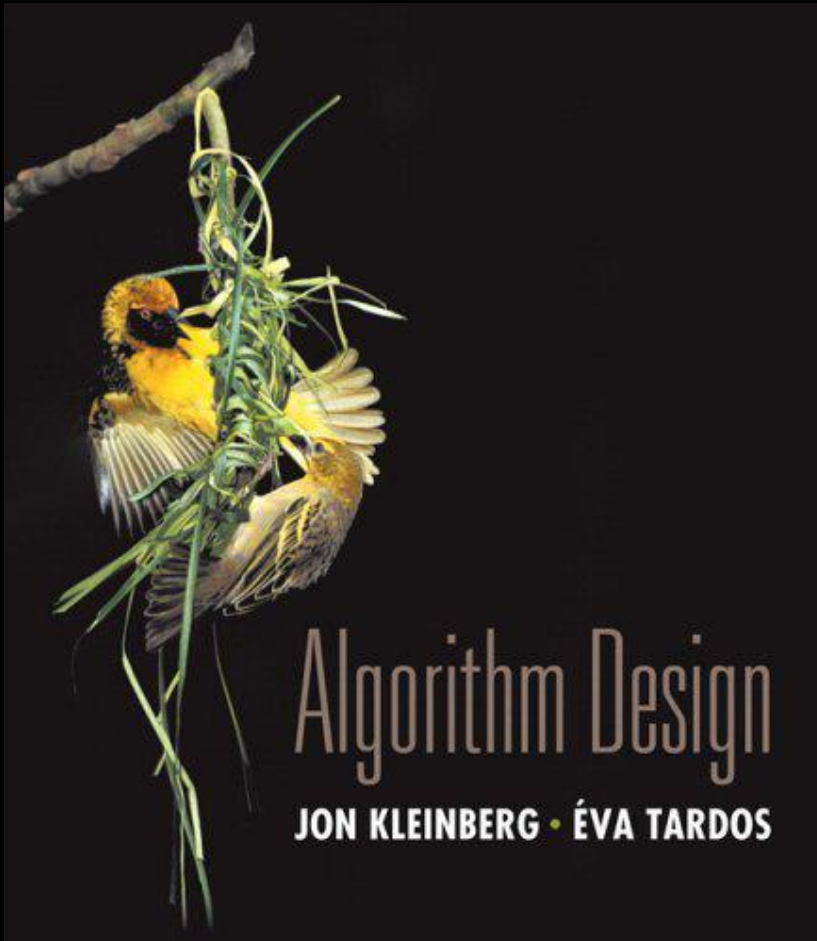


Chapter 8

NP and Computational Intractability



Slides by Kevin Wayne.
Copyright © 2005 Pearson-Addison Wesley.
All rights reserved.

8.3 Definition of NP

Decision Problems

Decision problem.

- X is a set of strings.
- Instance: string s .
- Algorithm A solves problem X : $A(s) = \text{yes}$ iff $s \in X$.

Polynomial time. Algorithm A runs in poly-time if for every string s , $A(s)$ terminates in at most $p(|s|)$ "steps", where $p(\cdot)$ is some polynomial.

PRIMES: $X = \{ 2, 3, 5, 7, 11, 13, 17, 23, 29, 31, 37, \dots \}$

Algorithm. [Agrawal-Kayal-Saxena, 2002] $p(|s|) = |s|^8$.

Definition of P

P. Decision problems for which there is a poly-time algorithm.

Problem	Description	Algorithm	Yes	No
MULTIPLE	Is x a multiple of y ?	Grade school division	51, 17	51, 16
RELPRIME	Are x and y relatively prime?	Euclid (300 BCE)	34, 39	34, 51
PRIMES	Is x prime?	AKS (2002)	53	51
EDIT-DISTANCE	Is the edit distance between x and y less than 5?	Dynamic programming	niether neither	acgggt ttttta
LSOLVE	Is there a vector x that satisfies $Ax = b$?	Gauss-Edmonds elimination	$\left[\begin{array}{ccc c} 0 & 1 & 1 & 4 \\ 2 & 4 & -2 & 2 \\ 0 & 3 & 15 & 36 \end{array} \right]$	$\left[\begin{array}{ccc c} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{array} \right]$

NP

Certification algorithm intuition.

- Certifier doesn't determine whether $s \in X$ on its own; rather, it checks a proposed proof t that $s \in X$.

Def. Algorithm $C(s, t)$ is a **certifier** for problem X if for every string s , $s \in X$ iff there exists a string t such that $C(s, t) = \text{yes}$.

NP. Decision problems for which there exists a **poly-time** certifier.

↑
 $C(s, t)$ is a poly-time algorithm and
 $|t| \leq p(|s|)$ for some polynomial $p(\cdot)$.

Remark. NP stands for **nondeterministic** polynomial-time.

Certifiers and Certificates: Composite

COMPOSITES. Given an integer s , is s composite?

Certificate. A nontrivial factor t of s . Note that such a certificate exists iff s is composite. Moreover $|t| \leq |s|$.

Certifier.

```
boolean C(s, t) {  
    if (t ≤ 1 or t ≥ s)  
        return false  
    else if (s is a multiple of t)  
        return true  
    else  
        return false  
}
```

Instance. $s = 437,669$.

Certificate. $t = 541$ or 809 . $\longleftarrow 437,669 = 541 \times 809$

Conclusion. COMPOSITES is in NP.

Certifiers and Certificates: 3-Satisfiability

SAT. Given a CNF formula Φ , is there a satisfying assignment?

Certificate. An assignment of truth values to the n boolean variables.

Certifier. Check that each clause in Φ has at least one true literal.

Ex.

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

instance s

$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1$$

certificate t

Conclusion. SAT is in NP.

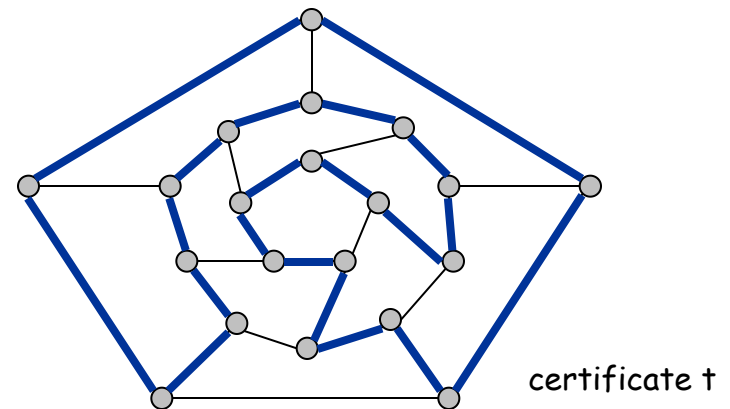
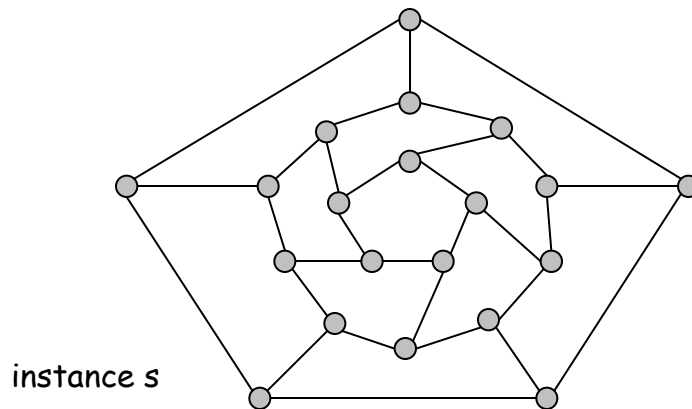
Certifiers and Certificates: Hamiltonian Cycle

HAM-CYCLE. Given an undirected graph $G = (V, E)$, does there exist a simple cycle C that visits every node?

Certificate. A permutation of the n nodes.

Certifier. Check that the permutation contains each node in V exactly once, and that there is an edge between each pair of adjacent nodes in the permutation.

Conclusion. HAM-CYCLE is in NP.



P, NP, EXP

P. Decision problems for which there is a **poly-time algorithm**.

EXP. Decision problems for which there is an **exponential-time algorithm**.

NP. Decision problems for which there is a **poly-time certifier**.

Claim. $P \subseteq NP$.

Pf. Consider any problem X in P .

- By definition, there exists a poly-time algorithm $A(s)$ that solves X .
- Certificate: $t = \varepsilon$, certifier $C(s, t) = A(s)$. ▪

Claim. $NP \subseteq EXP$.

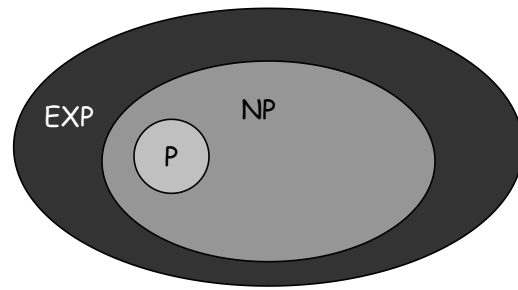
Pf. Consider any problem X in NP .

- By definition, there exists a poly-time certifier $C(s, t)$ for X .
- To solve input s , run $C(s, t)$ on all strings t with $|t| \leq p(|s|)$.
- Return **yes**, if $C(s, t)$ returns **yes** for any of these. ▪

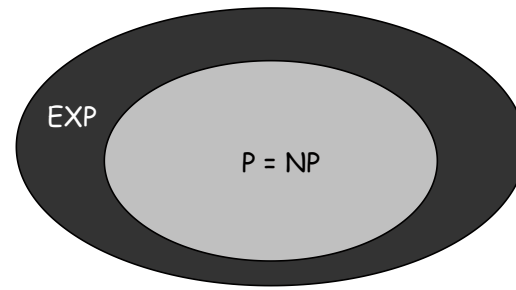
The Main Question: P Versus NP

Does $P = NP$? [Cook 1971, Edmonds, Levin, Yablonski, Gödel]

- Is the decision problem as easy as the certification problem?
- Clay \$1 million prize.



If $P \neq NP$



If $P = NP$

would break RSA cryptography
(and potentially collapse
economy)

If yes: Efficient algorithms for 3-COLOR, TSP, FACTOR, SAT, ...

If no: No efficient algorithms possible for 3-COLOR, TSP, SAT, ...

Consensus opinion on $P = NP$? Probably no.

8.4 NP-Completeness

Polynomial Transformation

Def. Problem X **polynomial reduces** (Cook) to problem Y if arbitrary instances of problem X can be solved using:

- Polynomial number of standard computational steps, plus
- Polynomial number of calls to oracle that solves problem Y .

Def. Problem X **polynomial transforms** (Karp) to problem Y if given any input x to X , we can construct an input y such that x is a yes instance of X iff y is a yes instance of Y .

↑
we require $|y|$ to be of size polynomial in $|x|$

Note. Polynomial transformation is polynomial reduction with just one call to oracle for Y , exactly at the end of the algorithm for X . Almost all previous reductions were of this form.

Open question. Are these two concepts the same?

NP-Complete

NP-complete. A problem Y in NP with the property that for every problem X in NP, $X \leq_p Y$.

Theorem. Suppose Y is an NP-complete problem. Then Y is solvable in poly-time iff $P = NP$.

Pf. \Leftarrow If $P = NP$ then Y can be solved in poly-time since Y is in NP.

Pf. \Rightarrow Suppose Y can be solved in poly-time.

- Let X be any problem in NP. Since $X \leq_p Y$, we can solve X in poly-time. This implies $NP \subseteq P$.
- We already know $P \subseteq NP$. Thus $P = NP$. ▪

Fundamental question. Do there exist "natural" NP-complete problems?

The "First" NP-Complete Problem

Theorem. SAT (3-SAT) is NP-complete. [Cook 1971, Levin 1973]

Establishing NP-Completeness

Recipe to establish NP-completeness of problem Y .

- Step 1. Show that Y is in NP.
- Step 2. Choose an NP-complete problem X .
- Step 3. Prove that $X \leq_p Y$.

Justification. If X is an NP-complete problem, and Y is a problem in NP with the property that $X \leq_p Y$ then Y is NP-complete.

Pf. Let W be any problem in NP. Then $W \leq_p X \leq_p Y$.

- By transitivity, $W \leq_p Y$.
- Hence Y is NP-complete. ▪

Some NP-Complete Problems

Six basic genres of NP-complete problems and paradigmatic examples.

- Packing problems: INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Numerical problems: SUBSET-SUM, PARTITION.

Practice. Most NP problems are either known to be in P or NP-complete.

Notable exceptions. Factoring, graph isomorphism, Nash equilibrium.

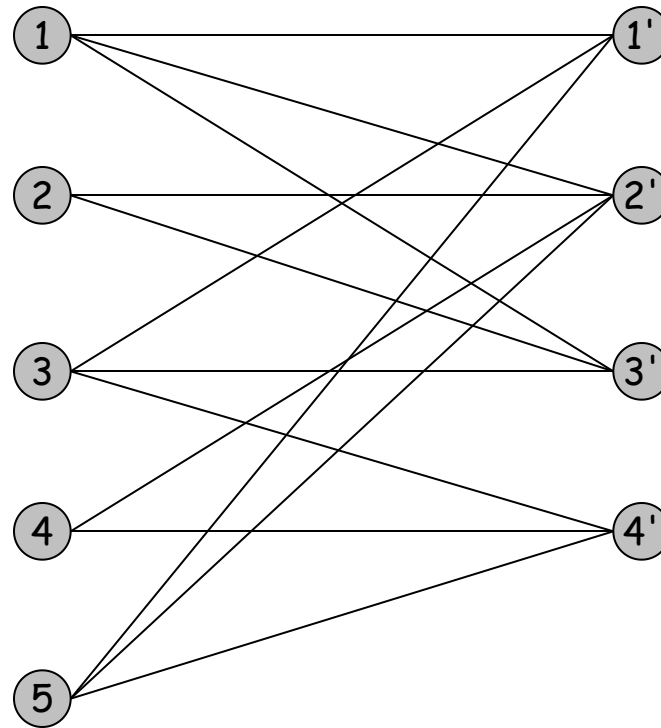
8.5 Sequencing Problems

Basic genres.

- Packing problems: INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- **Sequencing problems:** HAMILTONIAN-CYCLE, TSP.
- Numerical problems: SUBSET-SUM, PARTITION.

Hamiltonian Cycle

HAM-CYCLE: given an undirected graph $G = (V, E)$, does there exist a simple cycle Γ that contains every node in V .



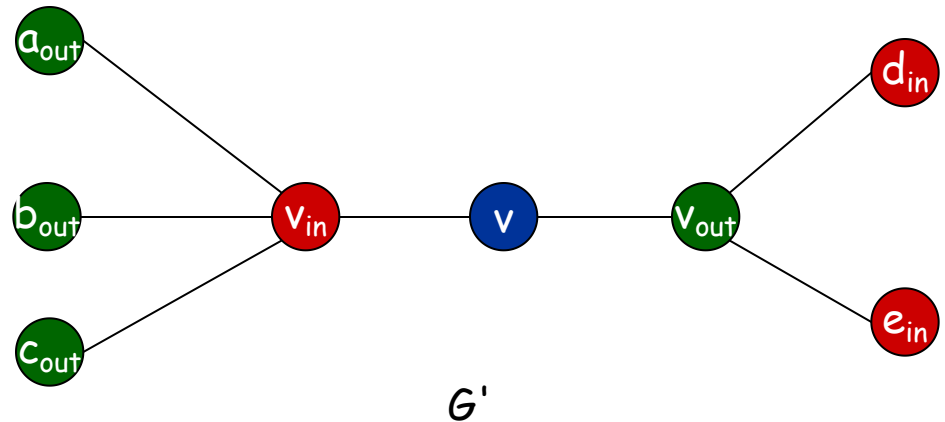
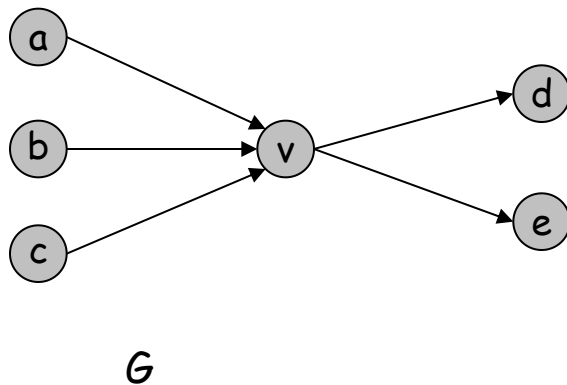
NO: bipartite graph with odd number of nodes.

Directed Hamiltonian Cycle

DIR-HAM-CYCLE: given a **digraph** $G = (V, E)$, does there exist a simple directed cycle Γ that contains every node in V ?

Claim. $\text{DIR-HAM-CYCLE} \leq_p \text{HAM-CYCLE}$.

Pf. Given a directed graph $G = (V, E)$, construct an undirected graph G' with $3n$ nodes.



Directed Hamiltonian Cycle

Claim. G has a Hamiltonian cycle iff G' does.

Pf. \Rightarrow

- Suppose G has a directed Hamiltonian cycle Γ .
- Then G' has an undirected Hamiltonian cycle (same order).

Pf. \Leftarrow

- Suppose G' has an undirected Hamiltonian cycle Γ' .
- Γ' must visit nodes in G' using one of following two orders:
 ..., B, G, R, B, G, R, B, G, R, B, ...
 ..., B, R, G, B, R, G, B, R, G, B, ...
- Blue nodes in Γ' make up directed Hamiltonian cycle Γ in G , or reverse of one. ▪

3-SAT Reduces to Directed Hamiltonian Cycle

Claim. $3\text{-SAT} \leq_p \text{DIR-HAM-CYCLE}$.

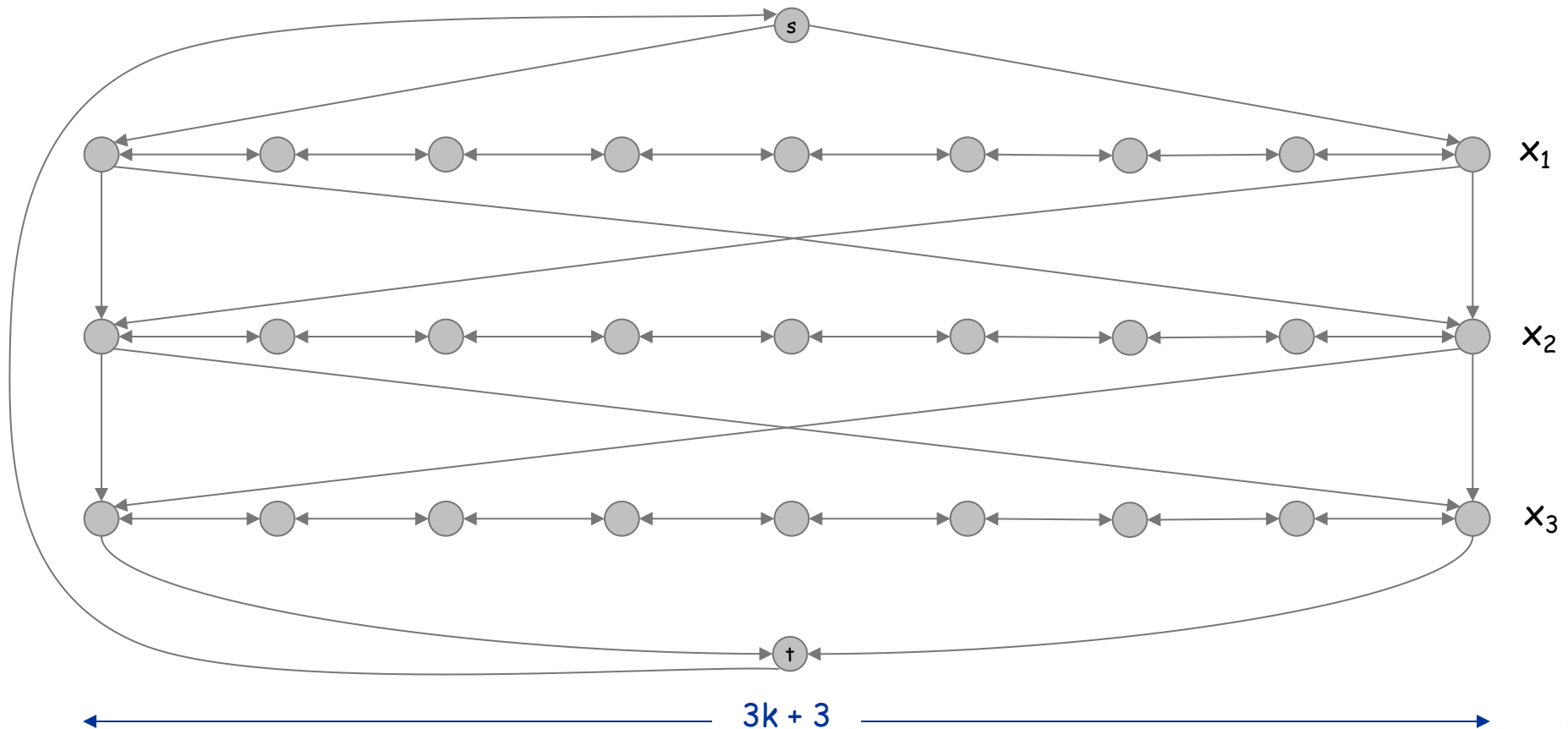
Pf. Given an instance Φ of 3-SAT, we construct an instance of DIR-HAM-CYCLE that has a Hamiltonian cycle iff Φ is satisfiable.

Construction. First, create graph that has 2^n Hamiltonian cycles which correspond in a natural way to 2^n possible truth assignments.

3-SAT Reduces to Directed Hamiltonian Cycle

Construction. Given 3-SAT instance Φ with n variables x_i and k clauses.

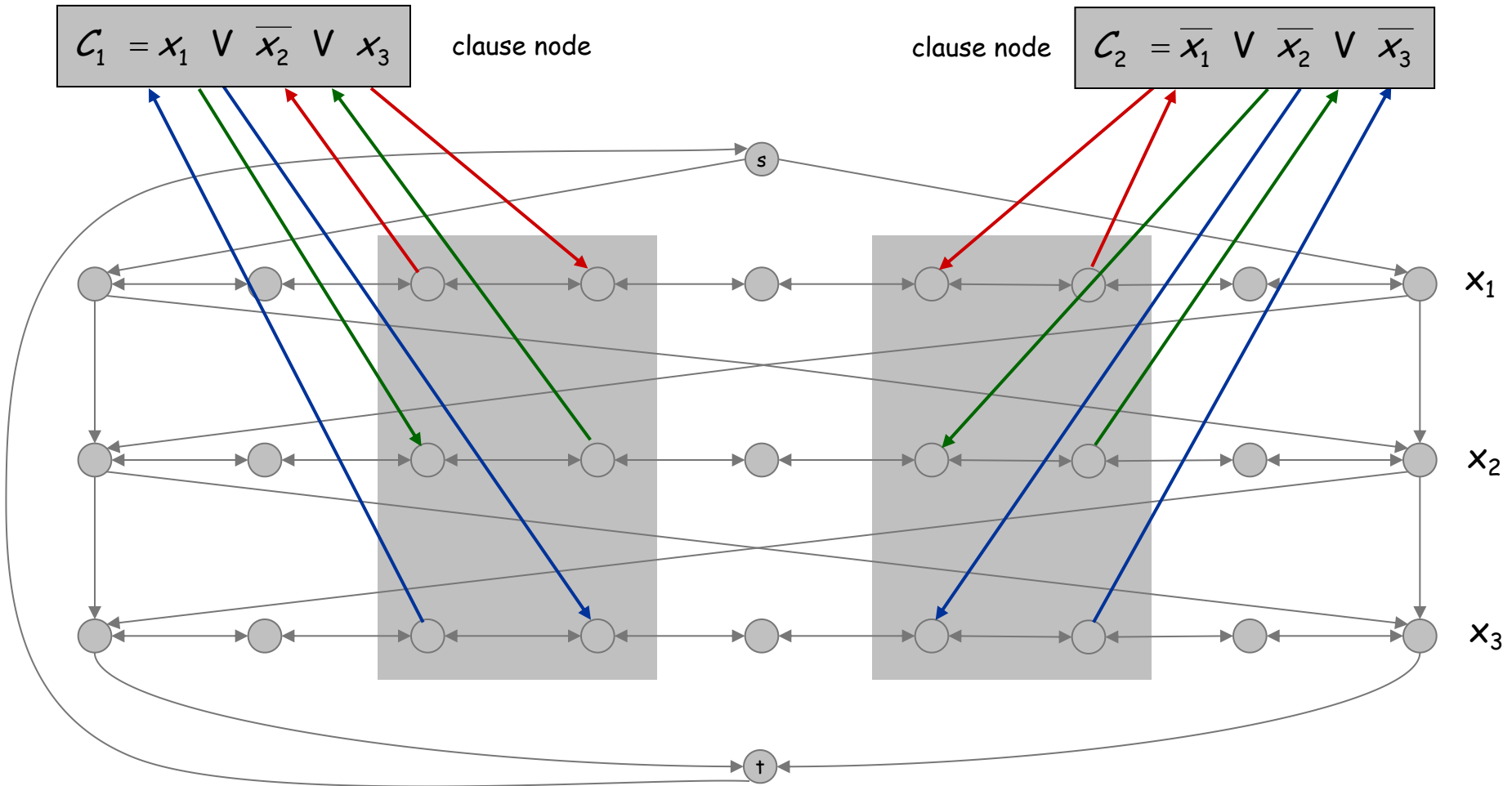
- Construct G to have 2^n Hamiltonian cycles.
- Intuition: traverse path i from left to right \Leftrightarrow set variable $x_i = 1$.



3-SAT Reduces to Directed Hamiltonian Cycle

Construction. Given 3-SAT instance Φ with n variables x_i and k clauses.

- For each clause: add a node and 6 edges.



3-SAT Reduces to Directed Hamiltonian Cycle

Claim. Φ is satisfiable iff G has a Hamiltonian cycle.

Pf. \Rightarrow

- Suppose 3-SAT instance has satisfying assignment x^* .
- Then, define Hamiltonian cycle in G as follows:
 - if $x_i^* = 1$, traverse row i from left to right
 - if $x_i^* = 0$, traverse row i from right to left
 - for each clause C_j , there will be at least one row i in which we are going in "correct" direction to splice node C_j into tour

3-SAT Reduces to Directed Hamiltonian Cycle

Claim. Φ is satisfiable iff G has a Hamiltonian cycle.

Pf. \Leftarrow

- Suppose G has a Hamiltonian cycle Γ .
- If Γ enters clause node C_j , it must depart on mate edge.
 - thus, nodes immediately before and after C_j are connected by an edge e in G
 - removing C_j from cycle, and replacing it with edge e yields Hamiltonian cycle on $G - \{C_j\}$
- Continuing in this way, we are left with Hamiltonian cycle Γ' in $G - \{C_1, C_2, \dots, C_k\}$.
- Set $x^*_i = 1$ iff Γ' traverses row i left to right.
- Since Γ visits each clause node C_j , at least one of the paths is traversed in "correct" direction, and each clause is satisfied. ▪

Traveling Salesperson Problem

TSP. Given a set of n cities and a pairwise distance function $d(u, v)$, is there a tour of length $\leq D$?

HAM-CYCLE: given a graph $G = (V, E)$, does there exist a simple cycle that contains every node in V ?

Claim. $\text{HAM-CYCLE} \leq_p \text{TSP}$.

Pf.

- Given instance $G = (V, E)$ of HAM-CYCLE, create n cities with distance function

$$d(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 2 & \text{if } (u, v) \notin E \end{cases}$$

- TSP instance has tour of length $\leq n$ iff G is Hamiltonian. ▪

Remark. TSP instance in reduction satisfies Δ -inequality.

8.8 Numerical Problems

Basic genres.

- Packing problems: INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Numerical problems: SUBSET-SUM, PARTITION.

Subset Sum

SUBSET-SUM. Given natural numbers w_1, \dots, w_n and an integer W , is there a subset that adds up to exactly W ?

Ex: $\{ 1, 4, 16, 64, 256, 1040, 1041, 1093, 1284, 1344 \}$, $W = 3754$.

Yes. $1 + 16 + 64 + 256 + 1040 + 1093 + 1284 = 3754$.

Remark. With arithmetic problems, input integers are encoded in binary. Polynomial reduction must be polynomial in **binary** encoding.

Claim. $3\text{-SAT} \leq_p \text{SUBSET-SUM}$.

Pf. Given an instance Φ of 3-SAT, we construct an instance of SUBSET-SUM that has solution iff Φ is satisfiable.

Subset Sum

Construction. Given 3-SAT instance Φ with n variables and k clauses, form $2n + 2k$ decimal integers, each of $n+k$ digits, as illustrated below.

Claim. Φ is satisfiable iff there exists a subset that sums to W .

Pf. No carries possible.

$$C_1 = \bar{x} \vee y \vee z$$

$$C_2 = x \vee \bar{y} \vee z$$

$$C_3 = \bar{x} \vee \bar{y} \vee \bar{z}$$

dummies to get
clause columns
to sum to 4

	x	y	z	C_1	C_2	C_3	
x	1	0	0	0	1	0	100,010
$\neg x$	1	0	0	1	0	1	100,101
y	0	1	0	1	0	0	10,100
$\neg y$	0	1	0	0	1	1	10,011
z	0	0	1	1	1	0	1,110
$\neg z$	0	0	1	0	0	1	1,001
}	0	0	0	1	0	0	100
	0	0	0	2	0	0	200
	0	0	0	0	1	0	10
	0	0	0	0	2	0	20
	0	0	0	0	0	1	1
	0	0	0	0	0	2	2
W	1	1	1	4	4	4	111,444

Partition

SUBSET-SUM. Given natural numbers w_1, \dots, w_n and an integer W , is there a subset that adds up to exactly W ?

PARTITION. Given natural numbers v_1, \dots, v_m , can they be partitioned into two subsets that add up to the same value?

$$\nwarrow \frac{1}{2} \sum_i v_i$$

Claim. $\text{SUBSET-SUM} \leq_p \text{PARTITION}$.

Pf. Let W, w_1, \dots, w_n be an instance of SUBSET-SUM.

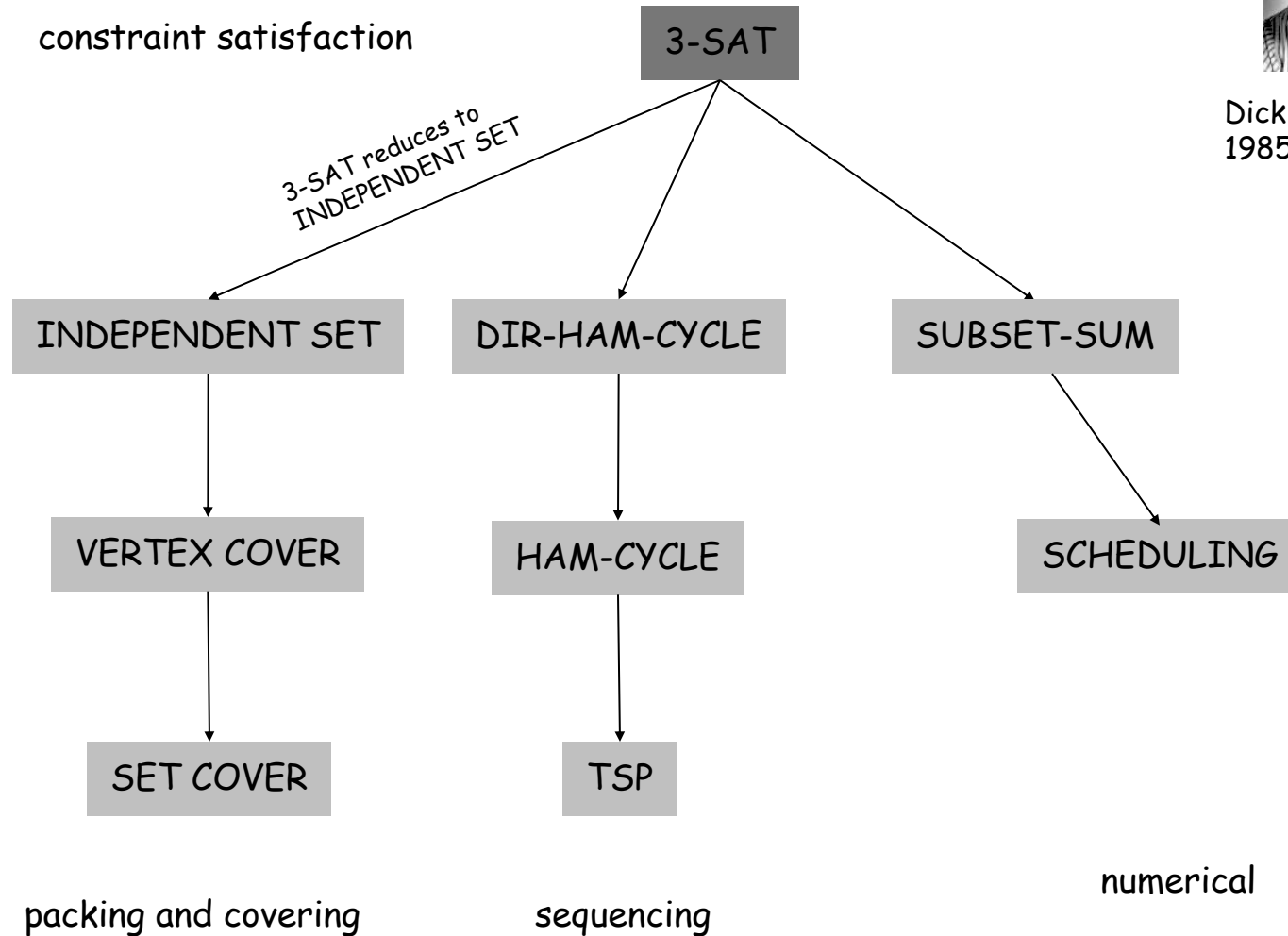
- Create instance of PARTITION with $m = n+2$ elements.
 - $v_1 = w_1, v_2 = w_2, \dots, v_n = w_n, v_{n+1} = 2 \sum_i w_i - W, v_{n+2} = \sum_i w_i + W$
- There exists a subset that sums to W iff there exists a partition since two new elements cannot be in the same partition. ▪

$v_{n+1} = 2 \sum_i w_i - W$	W	subset A
$v_{n+2} = \sum_i w_i + W$	$\sum_i w_i - W$	subset B

Polynomial-Time Reductions



Dick Karp (1972)
1985 Turing Award



8.9 co-NP and the Asymmetry of NP

Asymmetry of NP

Asymmetry of NP. We only need to have short proofs of *yes* instances.

Ex 1. SAT vs. TAUTOLOGY.

- Can prove a CNF formula is satisfiable by giving such an assignment.
- How could we prove that a formula is **not** satisfiable?

Ex 2. HAM-CYCLE vs. NO-HAM-CYCLE.

- Can prove a graph is Hamiltonian by giving such a Hamiltonian cycle.
- How could we prove that a graph is **not** Hamiltonian?

Remark. SAT is NP-complete, but how do we classify TAUTOLOGY?

NP and co-NP

NP. Decision problems for which there is a poly-time certifier.

Ex. SAT, HAM-CYCLE, COMPOSITES.

Def. Given a decision problem X , its **complement** \overline{X} is the same problem with the *yes* and *no* answers reverse.

Ex. $\overline{X} = \{ 0, 1, 4, 6, 8, 9, 10, 12, 14, 15, \dots \}$
 $X = \{ 2, 3, 5, 7, 11, 13, 17, 23, 29, \dots \}$

co-NP. Complements of decision problems in NP.

Ex. TAUTOLOGY, NO-HAM-CYCLE, PRIMES.

NP = co-NP ?

Fundamental question. Does $NP = co-NP$?

- Do yes instances have succinct certificates iff no instances do?
- Consensus opinion: no.

Theorem. If $NP \neq co-NP$, then $P \neq NP$.

Pf idea.

- P is closed under complementation.
- If $P = NP$, then NP is closed under complementation.
- In other words, $NP = co-NP$.
- This is the contrapositive of the theorem.