

Chapter 8

NP and Computational Intractability



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

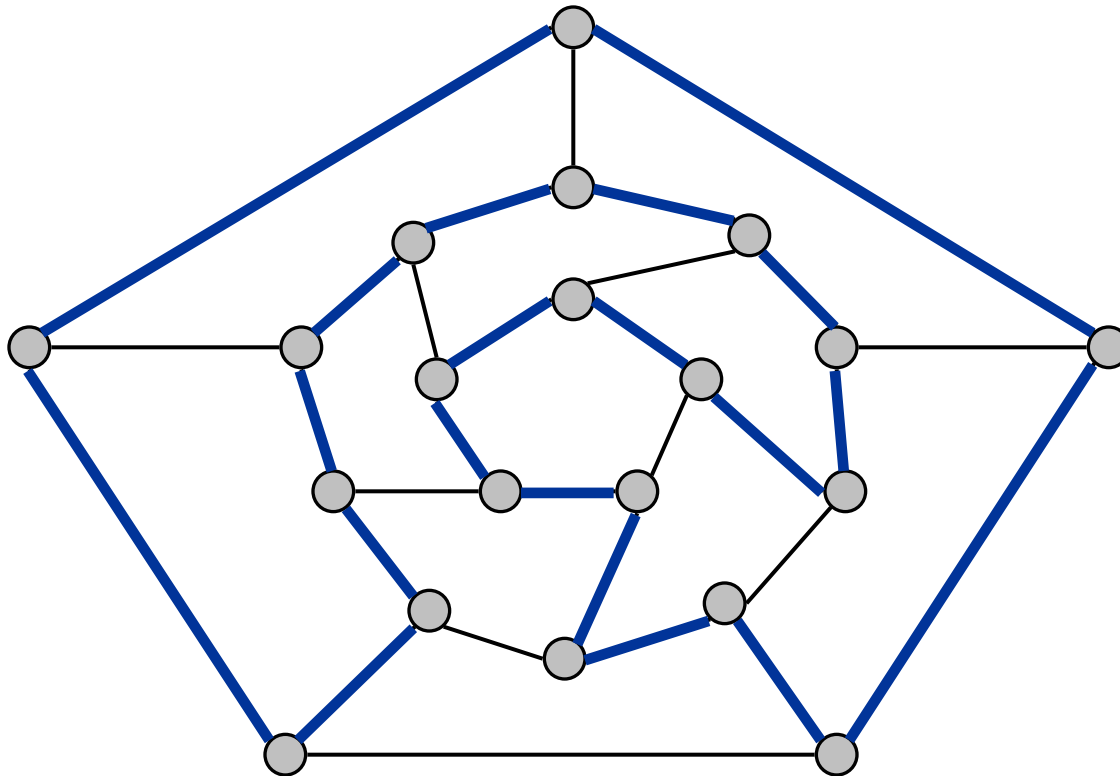
8.5 Sequencing Problems

Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- **Sequencing problems:** HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

Hamiltonian Cycle

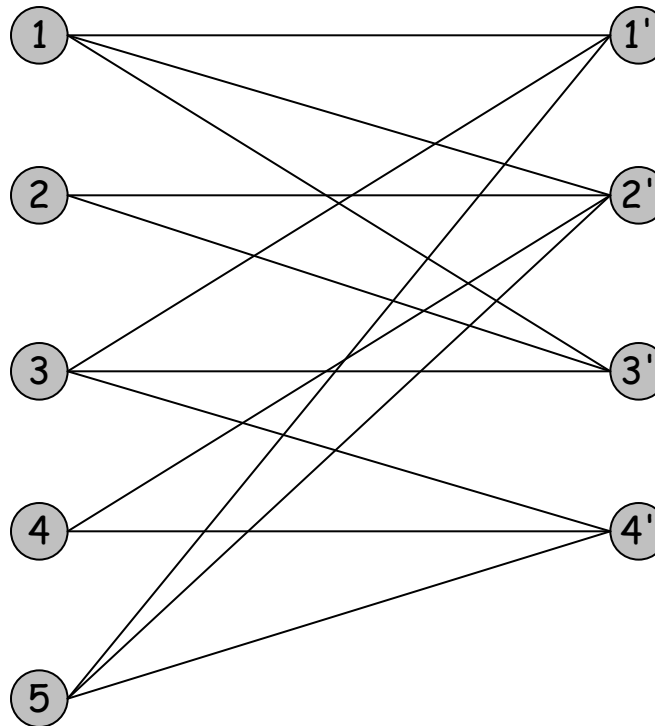
HAM-CYCLE: given an undirected graph $G = (V, E)$, does there exist a simple cycle Γ that contains every node in V .



YES: vertices and faces of a dodecahedron.

Hamiltonian Cycle

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



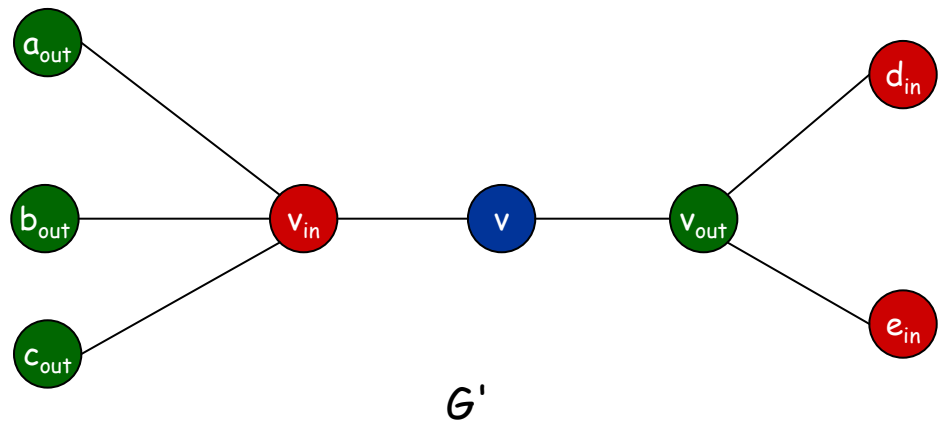
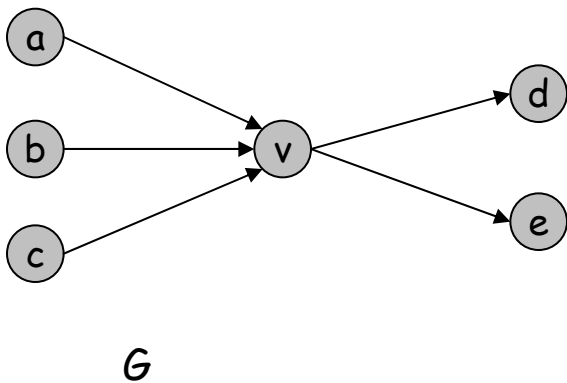
NO: bipartite graph with odd number of nodes.

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

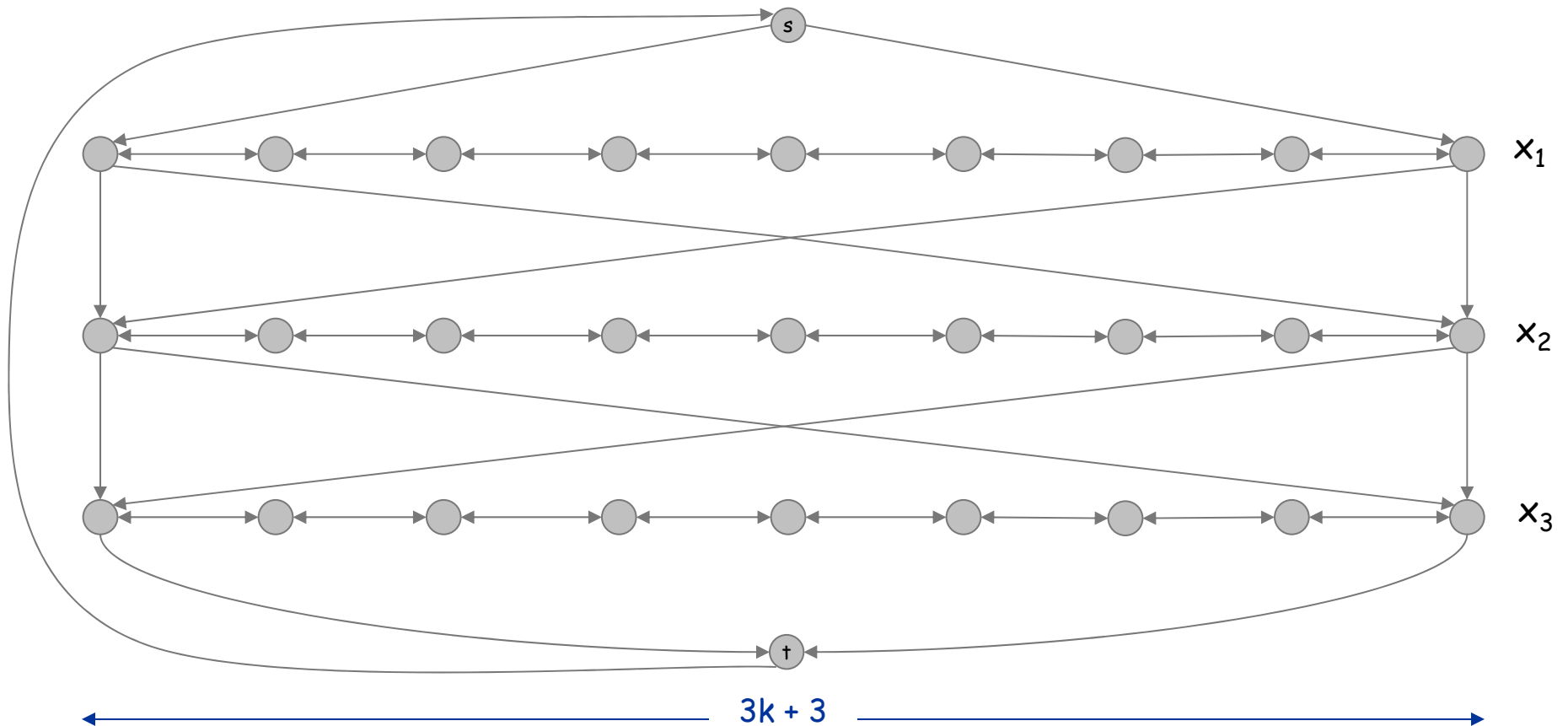
Pf. Given an instance Φ of 3-SAT, we construct an instance of DIR-HAM-CYCLE that has a Hamiltonian cycle iff Φ is satisfiable.

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.

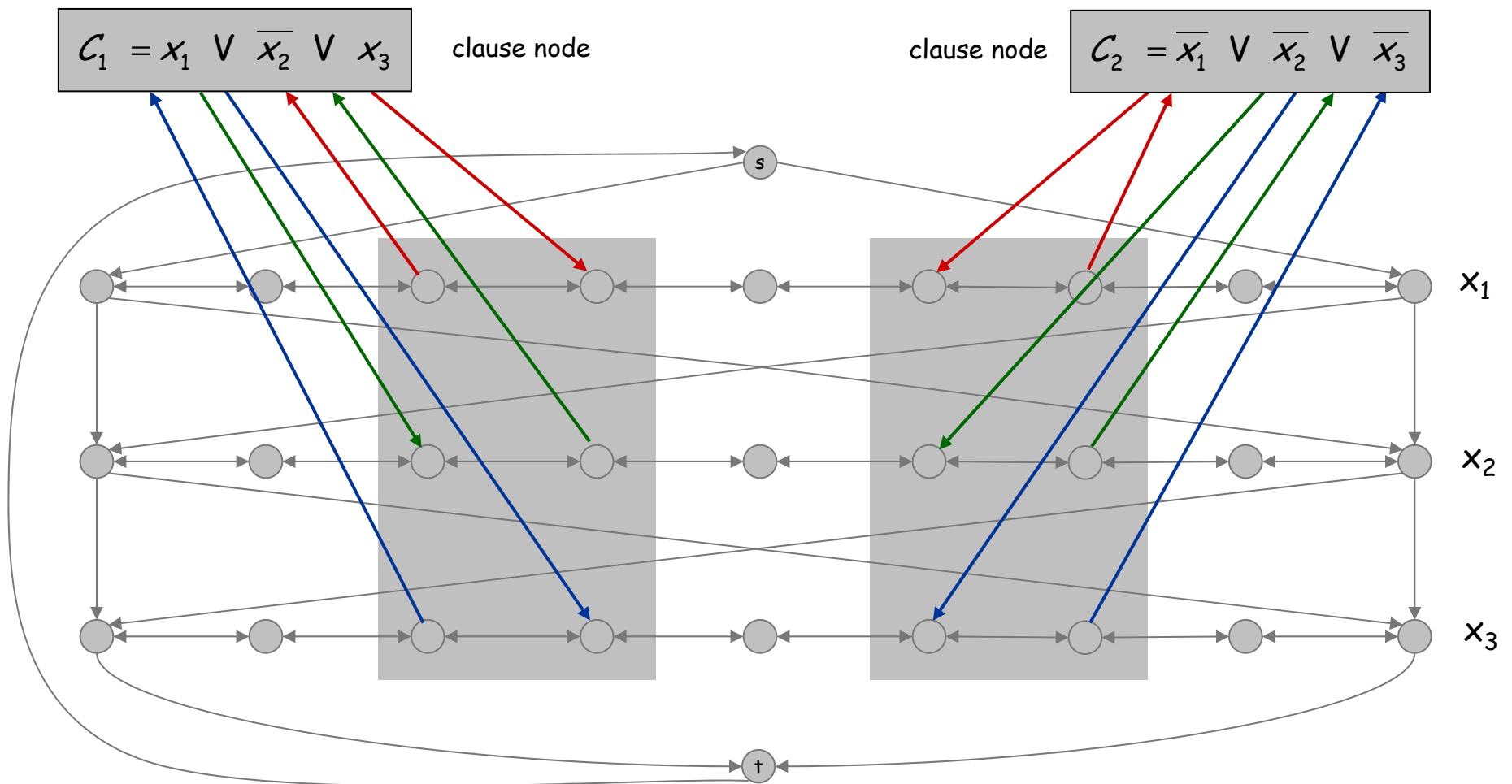
- Construct G to have 2^n Hamiltonian cycles.
- Intuition: traverse path i from left to right \Leftrightarrow set variable $x_i = 1$.



3-SAT Reduces to Directed Hamiltonian Cycle

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

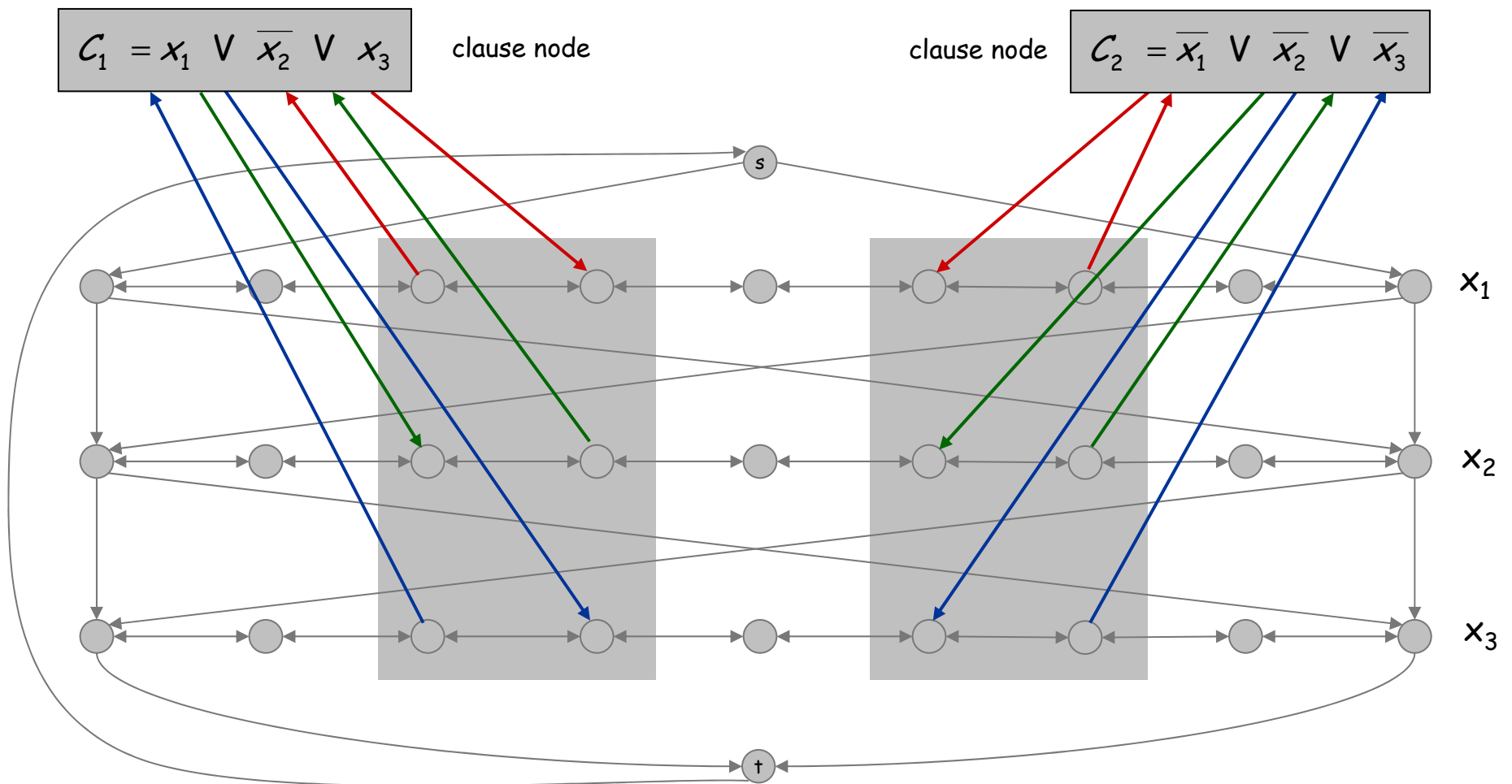
- For each clause: add a node and 6 edges.



3-SAT Reduces to Directed Hamiltonian Cycle

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

- If clause k contains literal z associated with x_i , then there is an edge from clause k to nodes $3k$ and $3k+1$ in the path corresponding to x_i



3-SAT Reduces to Directed Hamiltonian Cycle

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

- If clause k contains variable x_i there is an edge directed from node $(3k,i)$ to clause k and an edge from clause k to node $(3k+1,i)$
- If clause k contains literal NOT x_i there is an edge directed from node $(3k+1,i)$ to clause k and an edge from clause k to node $(3k,i)$

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. ■

Longest Path

SHORTEST-PATH. Given a digraph $G = (V, E)$, does there exist a simple path of length **at most** k edges?

LONGEST-PATH. Given a digraph $G = (V, E)$, does there exist a simple path of length **at least** k edges?

Claim. $3\text{-SAT} \leq_p \text{LONGEST-PATH}$.

Pf 1. Redo proof for DIR-HAM-CYCLE , ignoring back-edge from t to s .

Pf 2. Show $\text{HAM-CYCLE} \leq_p \text{LONGEST-PATH}$.

The Longest Path †

Lyrics. Copyright © 1988 by Daniel J. Barrett.

Music. Sung to the tune of *The Longest Time* by Billy Joel.



Woh-oh-oh-oh, find the longest path!
Woh-oh-oh-oh, find the longest path!

If you said P is NP tonight,
There would still be papers left to write,
I have a weakness,
I'm addicted to completeness,
And I keep searching for the longest path.

The algorithm I would like to see
Is of polynomial degree,
But it's elusive:
Nobody has found conclusive
Evidence that we can find a longest path.

I have been hard working for so long.
I swear it's right, and he marks it wrong.
Some how I'll feel sorry when it's done:
GPA 2.1
Is more than I hope for.

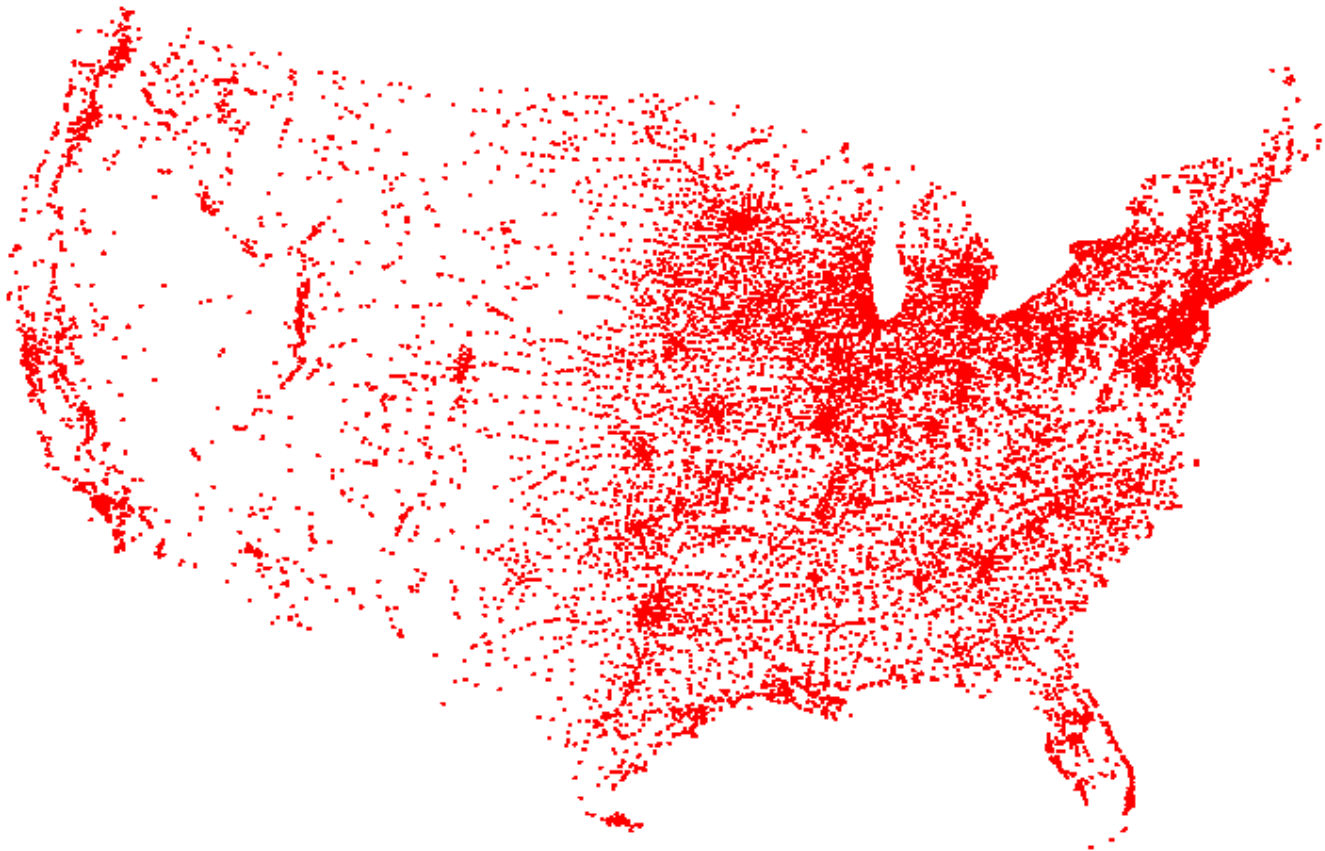
Garey, Johnson, Karp and other men (and women)
Tried to make it order $N \log N$.
Am I a mad fool
If I spend my life in grad school,
Forever following the longest path?

Woh-oh-oh-oh, find the longest path!
Woh-oh-oh-oh, find the longest path!
Woh-oh-oh-oh, find the longest path.

† Recorded by Dan Barrett while a grad student at Johns Hopkins during a difficult algorithms final.

Traveling Salesperson Problem

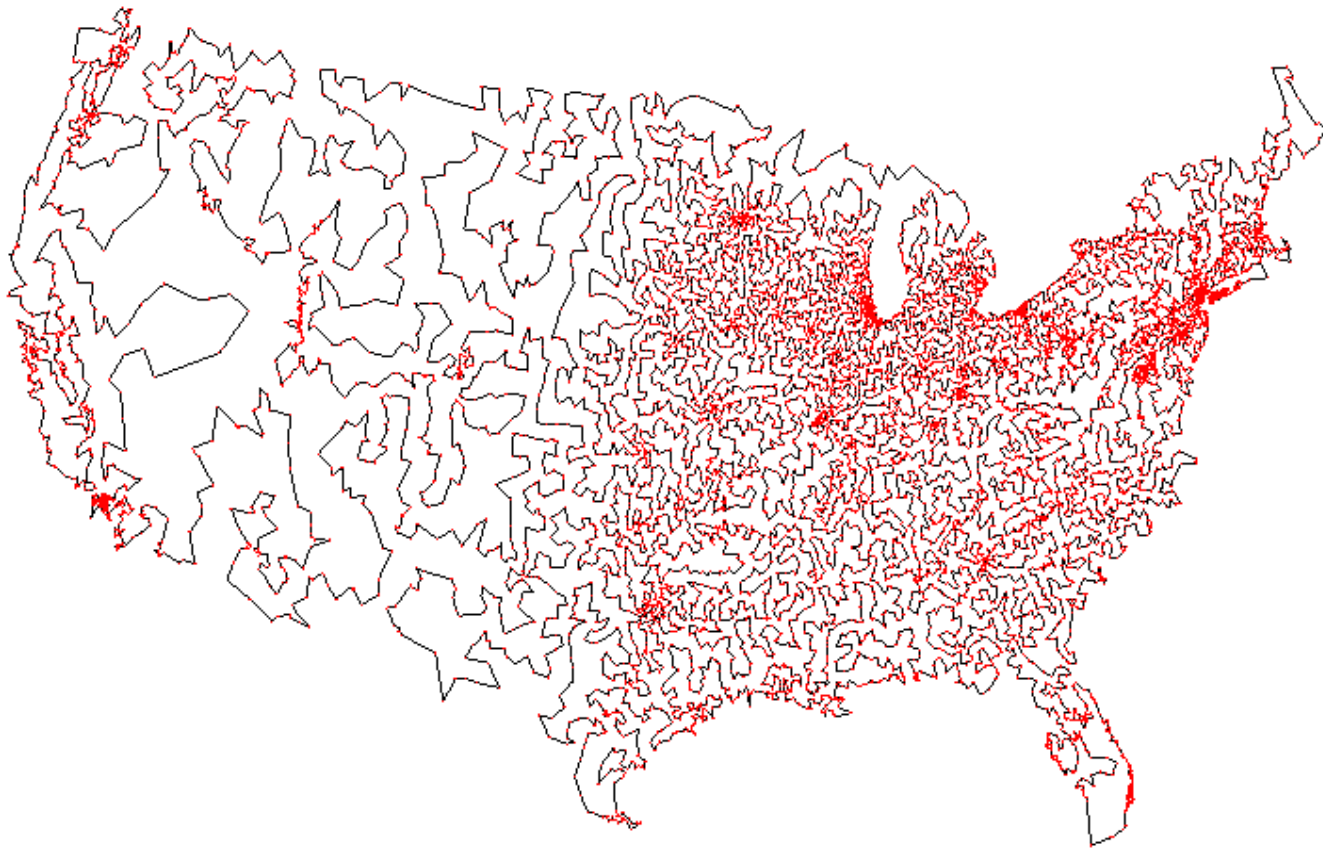
TSP. Given a set of n cities and a pairwise distance function $d(u, v)$, is there a tour of length $\leq D$?



All 13,509 cities in US with a population of at least 500
Reference: <http://www.tsp.gatech.edu>

Traveling Salesperson Problem

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



Optimal TSP tour
Reference: <http://www.tsp.gatech.edu>

Traveling Salesperson Problem

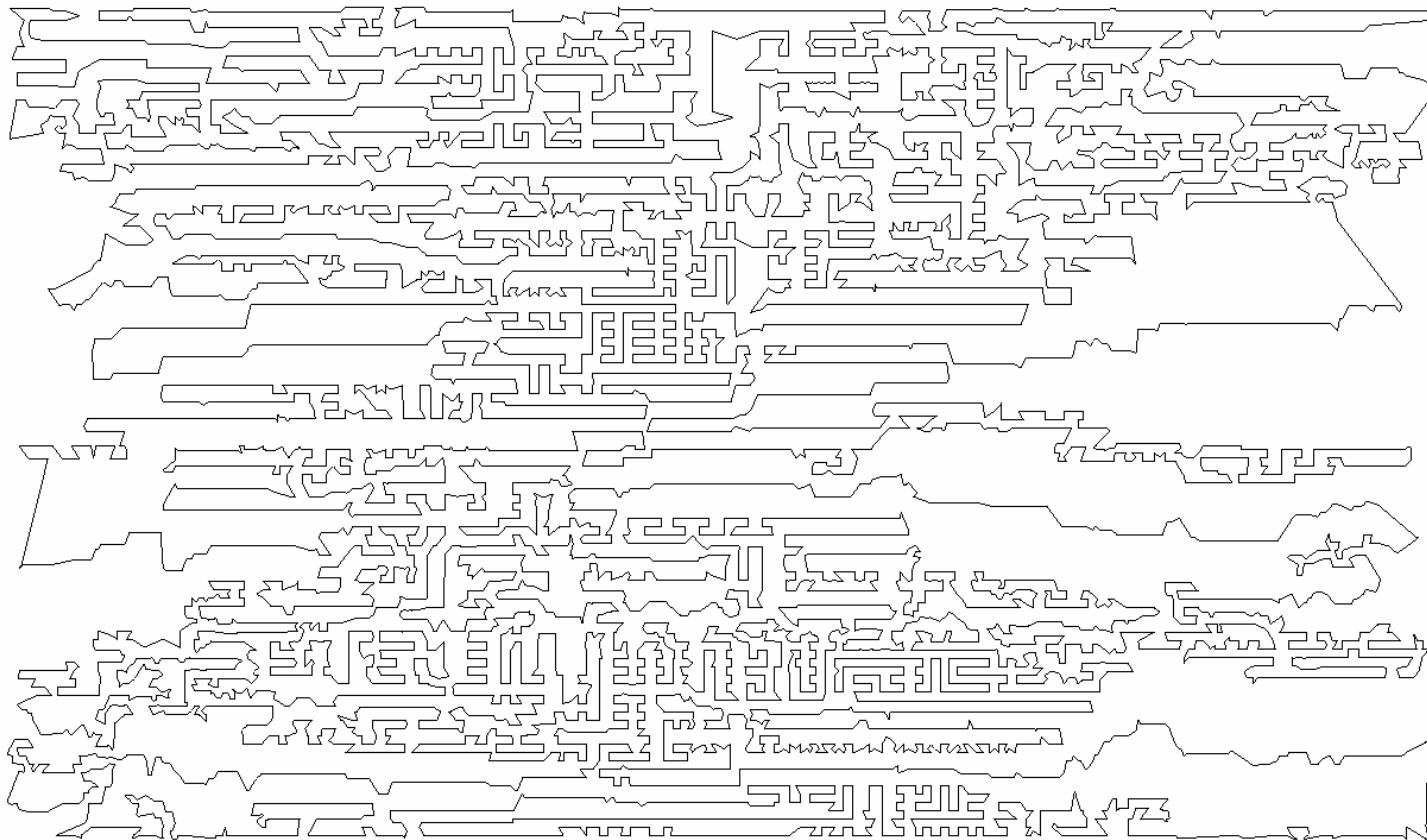
TSP. Given a set of n cities and a pairwise distance function $d(u, v)$, is there a tour of length $\leq D$?



11,849 holes to drill in a programmed logic array
Reference: <http://www.tsp.gatech.edu>

Traveling Salesperson Problem

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



Optimal TSP tour
Reference: <http://www.tsp.gatech.edu>

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.

8.6 Partitioning Problems

Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- **Partitioning problems:** 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

3-Dimensional Matching

3D-MATCHING. Given n instructors, n courses, and n times, and a list of the possible courses and times each instructor is willing to teach, is it possible to make an assignment so that all courses are taught at different times?

Instructor	Course	Time
Wayne	COS 423	MW 11-12:20
Wayne	COS 423	TTh 11-12:20
Wayne	COS 226	TTh 11-12:20
Wayne	COS 126	TTh 11-12:20
Tardos	COS 523	TTh 3-4:20
Tardos	COS 423	TTh 11-12:20
Tardos	COS 423	TTh 3-4:20
Kleinberg	COS 226	TTh 3-4:20
Kleinberg	COS 226	MW 11-12:20
Kleinberg	COS 423	MW 11-12:20

3-Dimensional Matching

3D-MATCHING. Given disjoint sets X , Y , and Z , each of size n and a set $T \subseteq X \times Y \times Z$ of triples, does there exist a set of n triples in T such that each element of $X \cup Y \cup Z$ is in exactly one of these triples?

Claim. $3\text{-SAT} \leq_p 3\text{D-MATCHING}$

Pf. Given an instance Φ of 3-SAT, we construct an instance of 3D-matching that has a perfect matching iff Φ is satisfiable.

3-Dimensional Matching

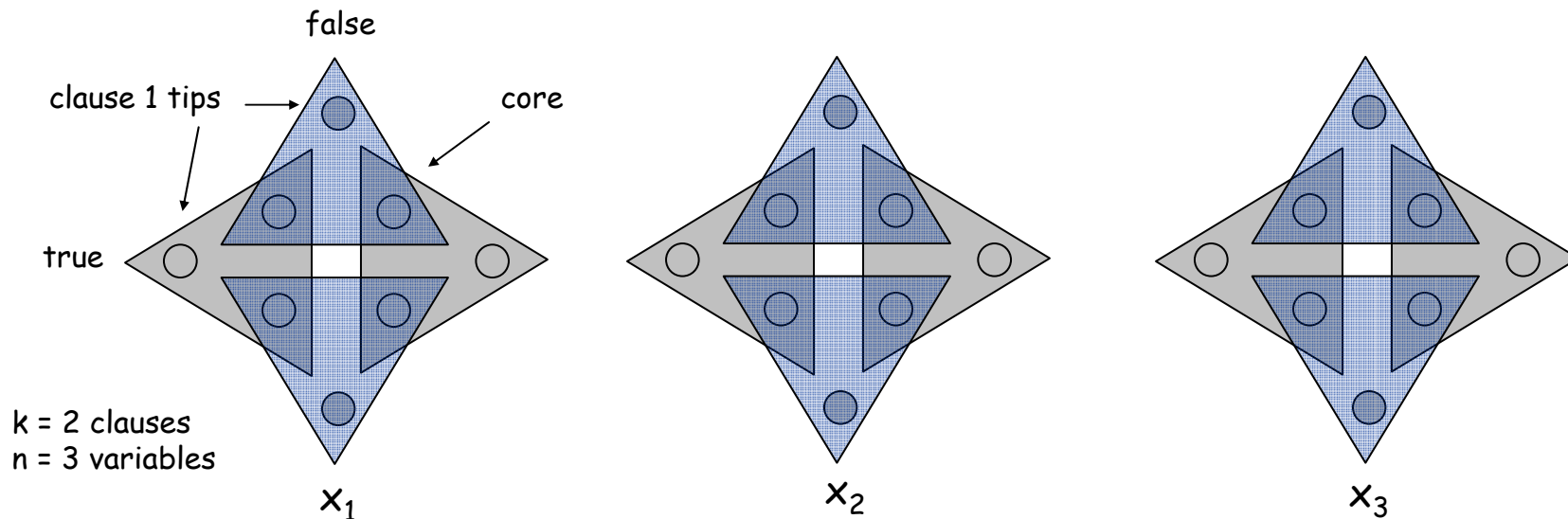
Construction. (part 1)

- Create gadget for each variable x_i with $2k$ core and tip elements.
- No other triples will use core elements.
- In gadget i , 3D-matching must use either both grey triples or both blue ones.

number of clauses

↑
set $x_i = \text{true}$

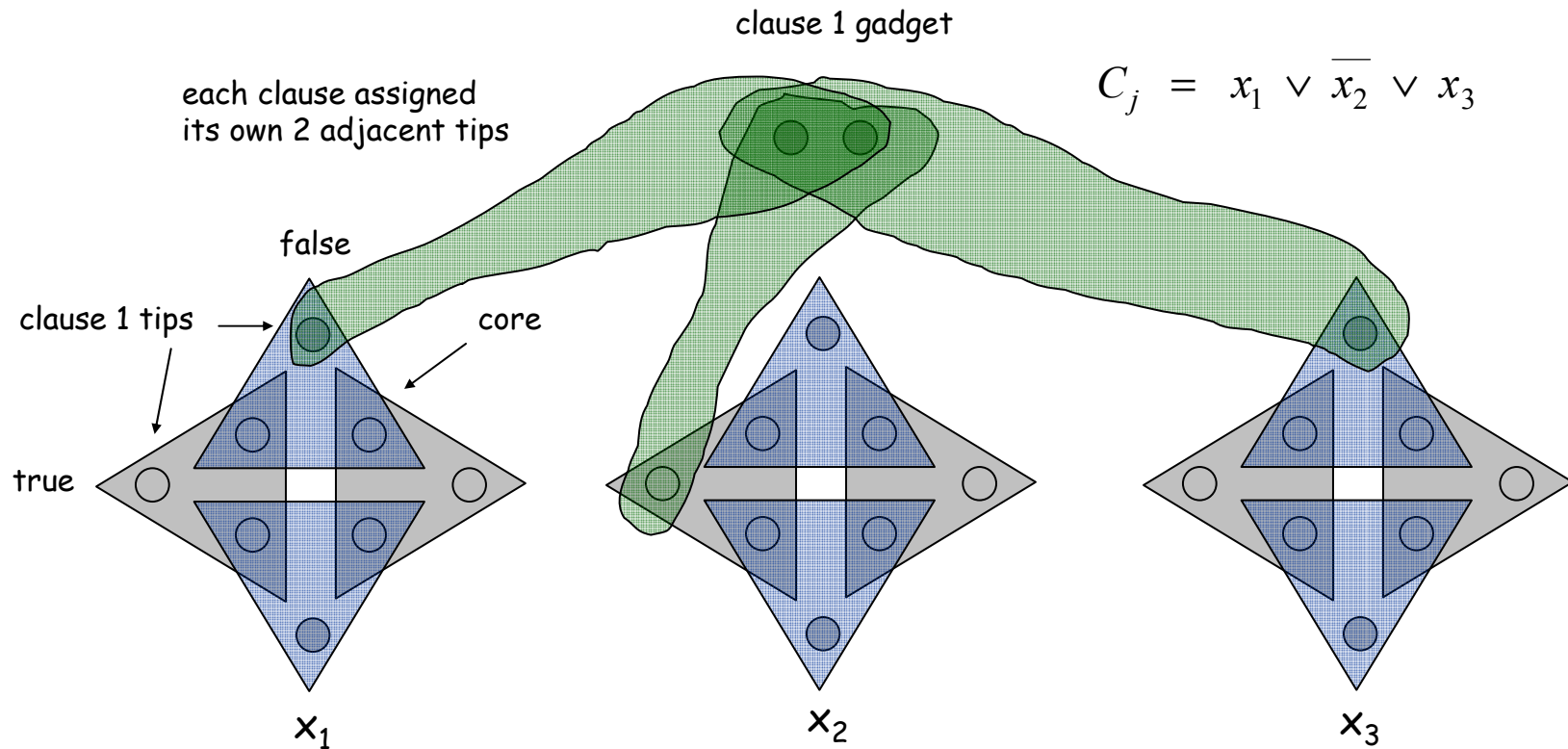
↑
set $x_i = \text{false}$



3-Dimensional Matching

Construction. (part 2)

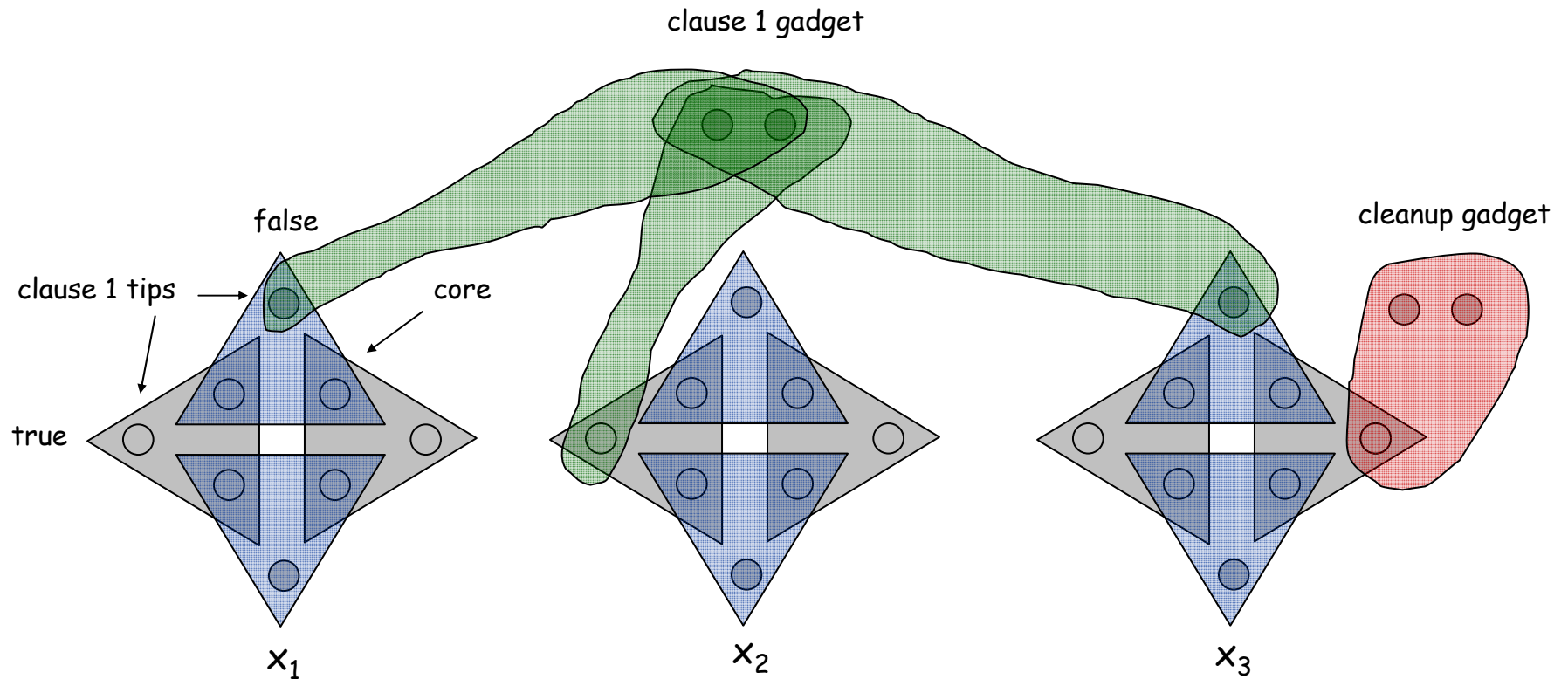
- For each clause C_j create two elements and three triples.
- Exactly one of these triples will be used in any 3D-matching.
- Ensures any 3D-matching uses either (i) grey core of x_1 or (ii) blue core of x_2 or (iii) grey core of x_3 .



3-Dimensional Matching

Construction. (part 3)

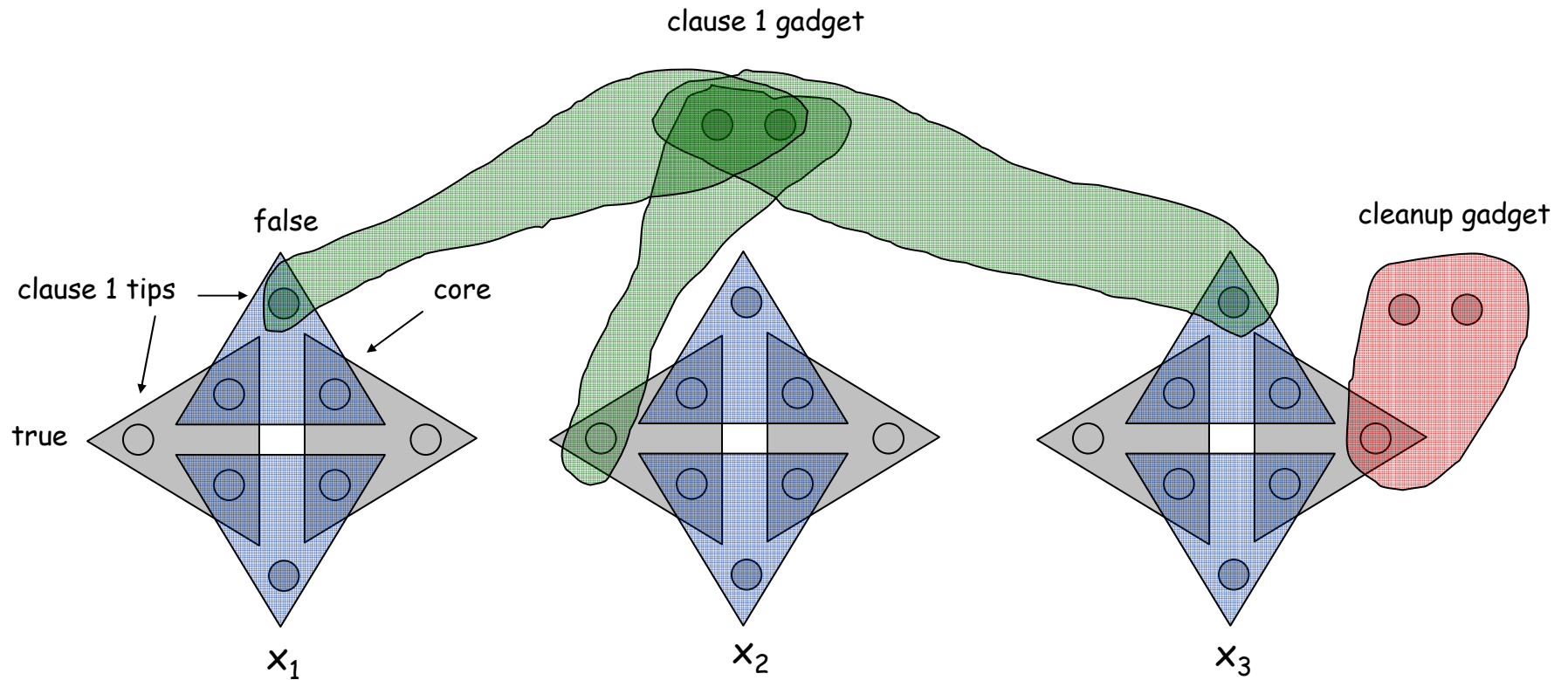
- Add $(n-1)k$ cleanup gadgets
- For each new gadget i , add $2nk$ triples: every triple share two common elements $\{q(i), q'(i)\}$. The third element is associated with each one of the $2nk$ existing tips



3-Dimensional Matching

Claim. Instance has a 3D-matching iff Φ is satisfiable.

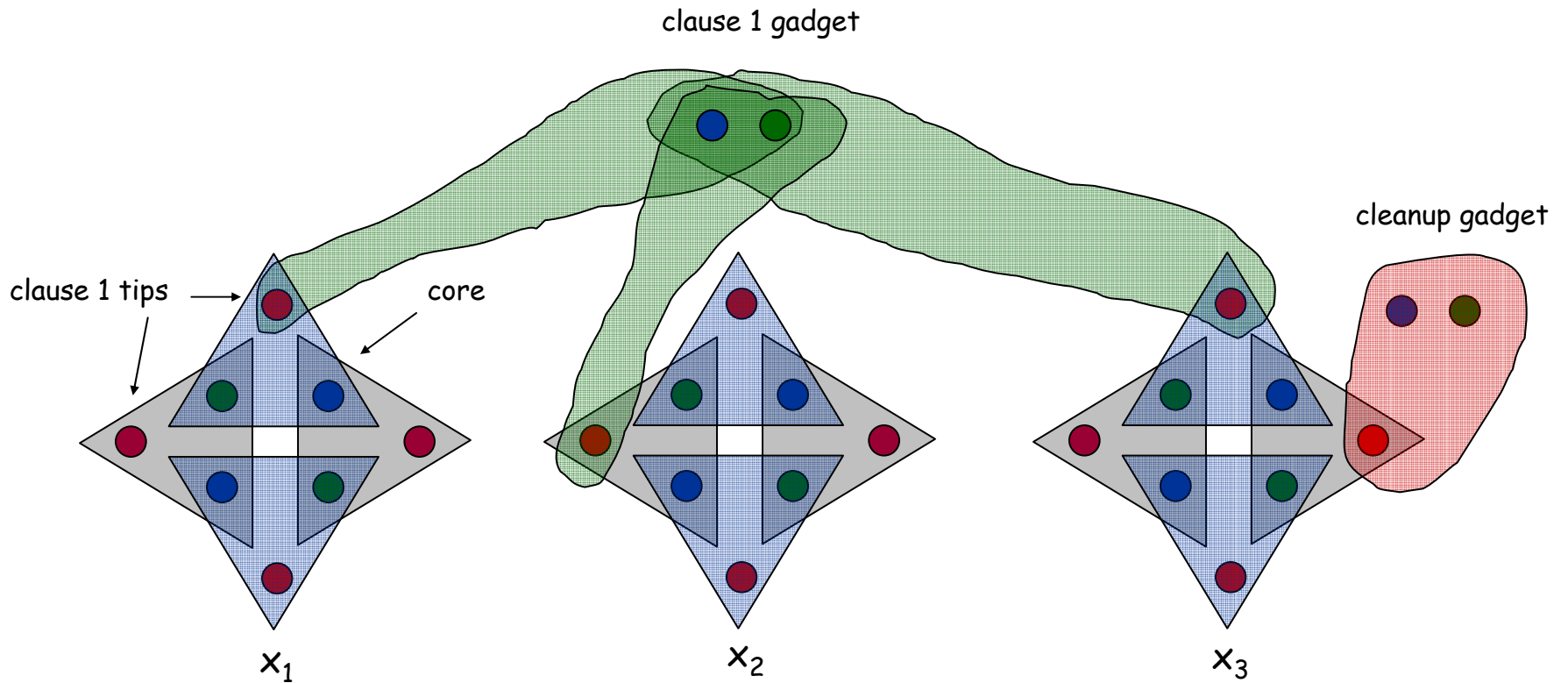
Detail. What are X , Y , and Z ? Does each triple contain one element from each of X , Y , Z ?



3-Dimensional Matching

Claim. Instance has a 3D-matching iff Φ is satisfiable.

Detail. What are X , Y , and Z ? Does each triple contain one element from each of X , Y , Z ?



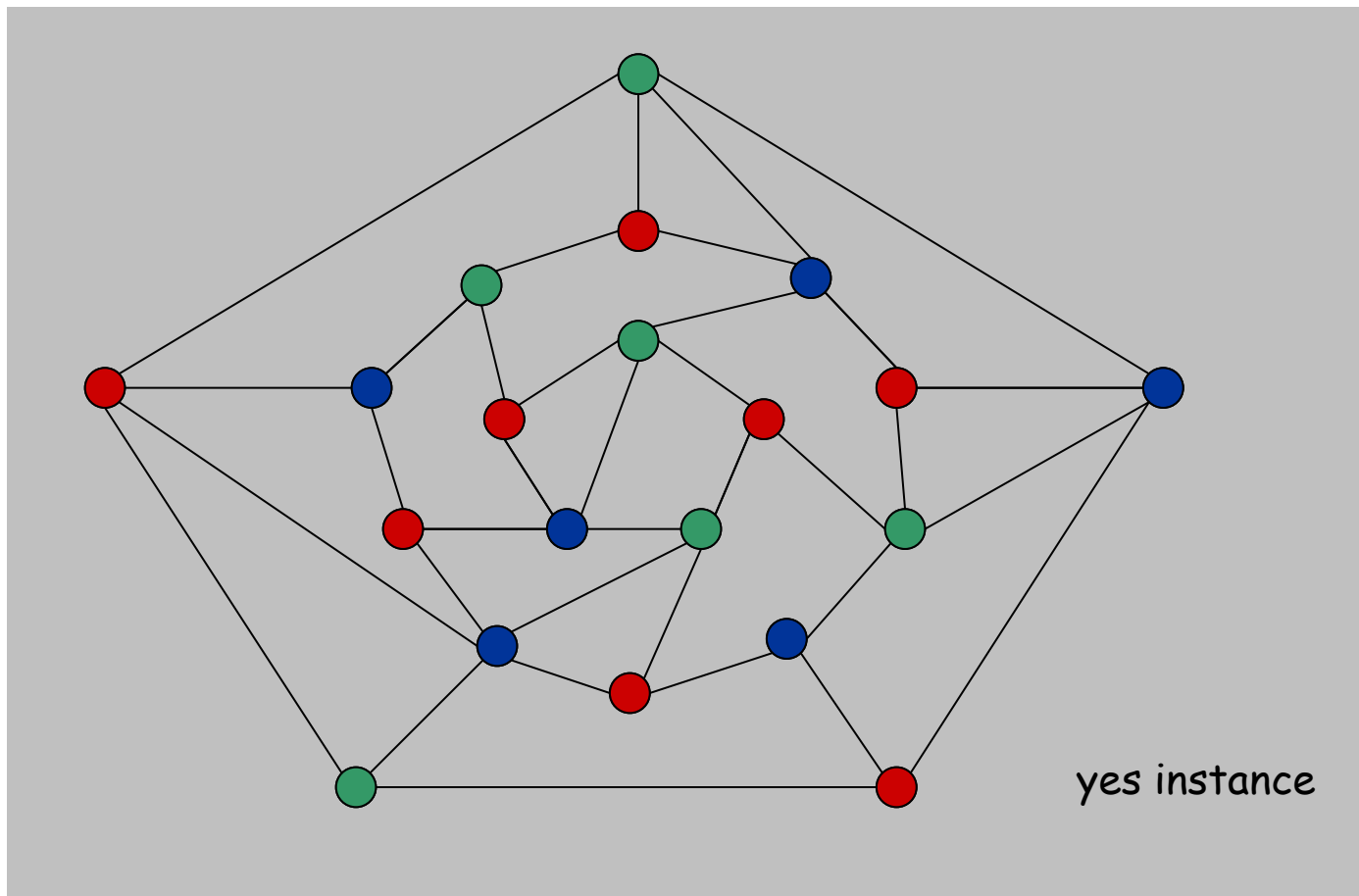
8.7 Graph Coloring

Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- **Partitioning problems:** 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

3-Colorability

3-COLOR: Given an undirected graph G does there exist a way to color the nodes red, green, and blue so that no adjacent nodes have the same color?



Register Allocation

Register allocation. Assign program variables to machine register so that no more than k registers are used and no two program variables that are needed at the same time are assigned to the same register.

Interference graph. Nodes are program variables names, edge between u and v if there exists an operation where both u and v are "live" at the same time.

Observation. [Chaitin 1982] Can solve register allocation problem iff interference graph is k -colorable.

Fact. $3\text{-COLOR} \leq_p k\text{-REGISTER-ALLOCATION}$ for any constant $k \geq 3$.

3-Colorability

Claim. $3\text{-SAT} \leq_p 3\text{-COLOR}$.

Pf. Given 3-SAT instance Φ , we construct an instance of 3-COLOR that is 3-colorable iff Φ is satisfiable.

Construction.

- i. For each literal, create a node.
- ii. Create 3 new nodes T, F, B; connect them in a triangle, and connect each literal to B.
- iii. Connect each literal to its negation.
- iv. For each clause, add gadget of 6 nodes and 13 edges.

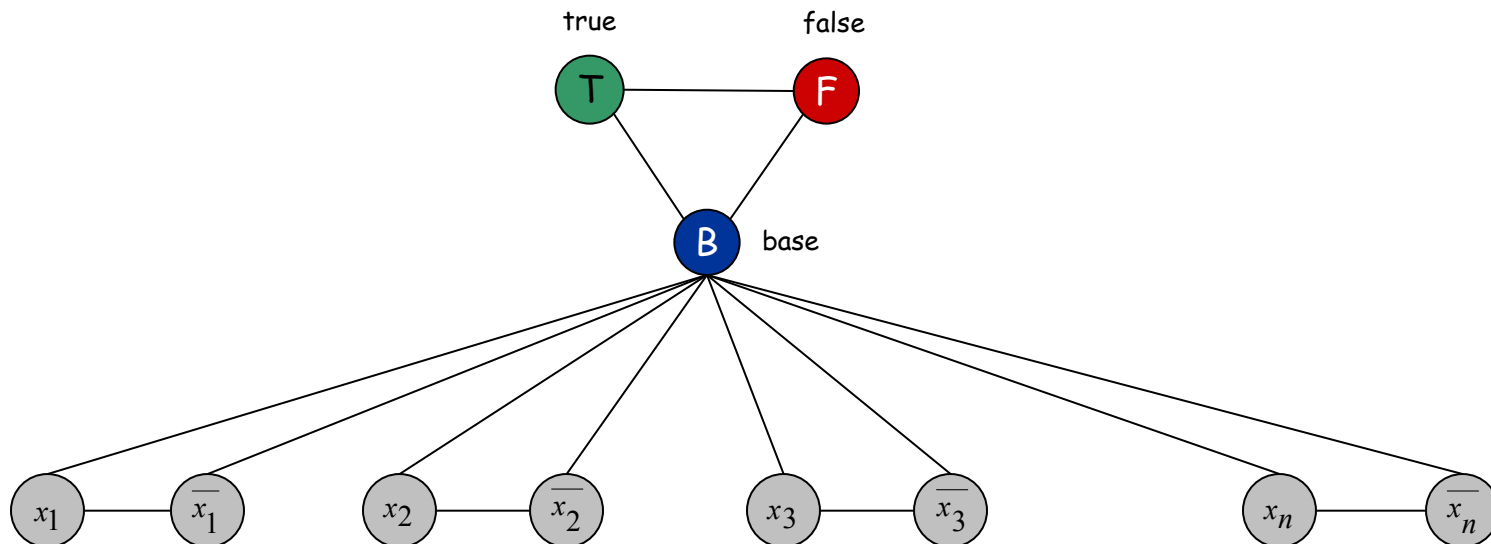
↑
to be described next

3-Colorability

Claim. Graph is 3-colorable iff Φ is satisfiable.

Pf. \Rightarrow Suppose graph is 3-colorable.

- Consider assignment that sets all T literals to true.
- (ii) ensures each literal is T or F.
- (iii) ensures a literal and its negation are opposites.

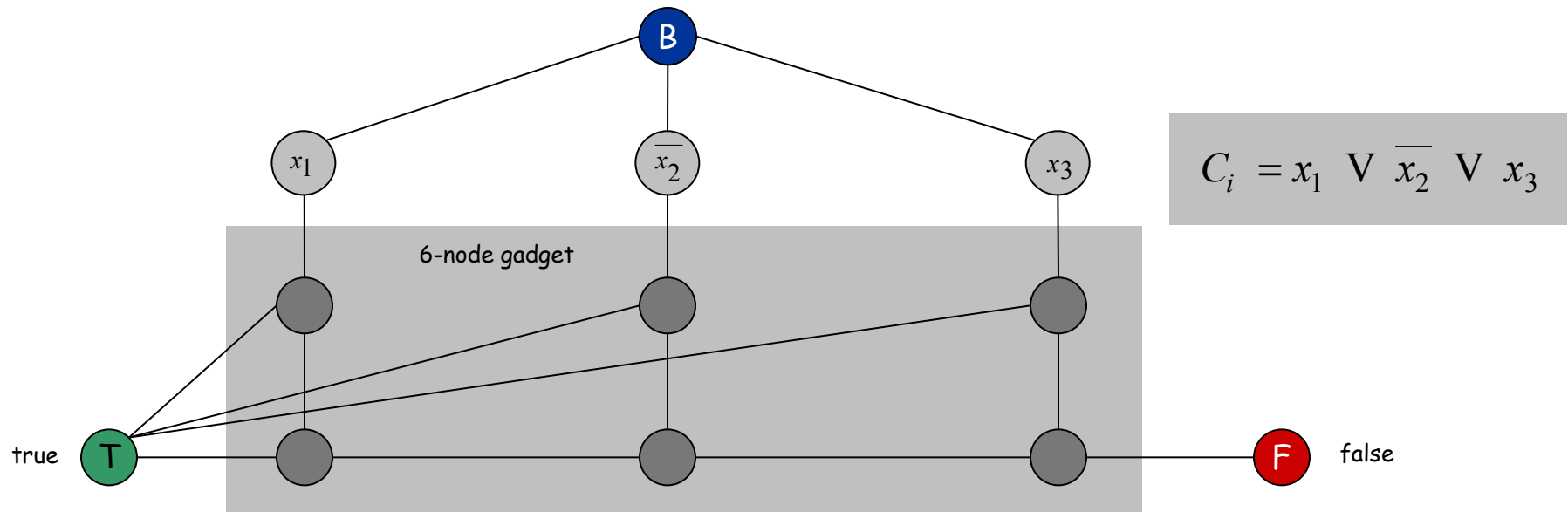


3-Colorability

Claim. Graph is 3-colorable iff Φ is satisfiable.

Pf. \Rightarrow Suppose graph is 3-colorable.

- Consider assignment that sets all T literals to true.
- (ii) ensures each literal is T or F.
- (iii) ensures a literal and its negation are opposites.
- (iv) ensures at least one literal in each clause is T.

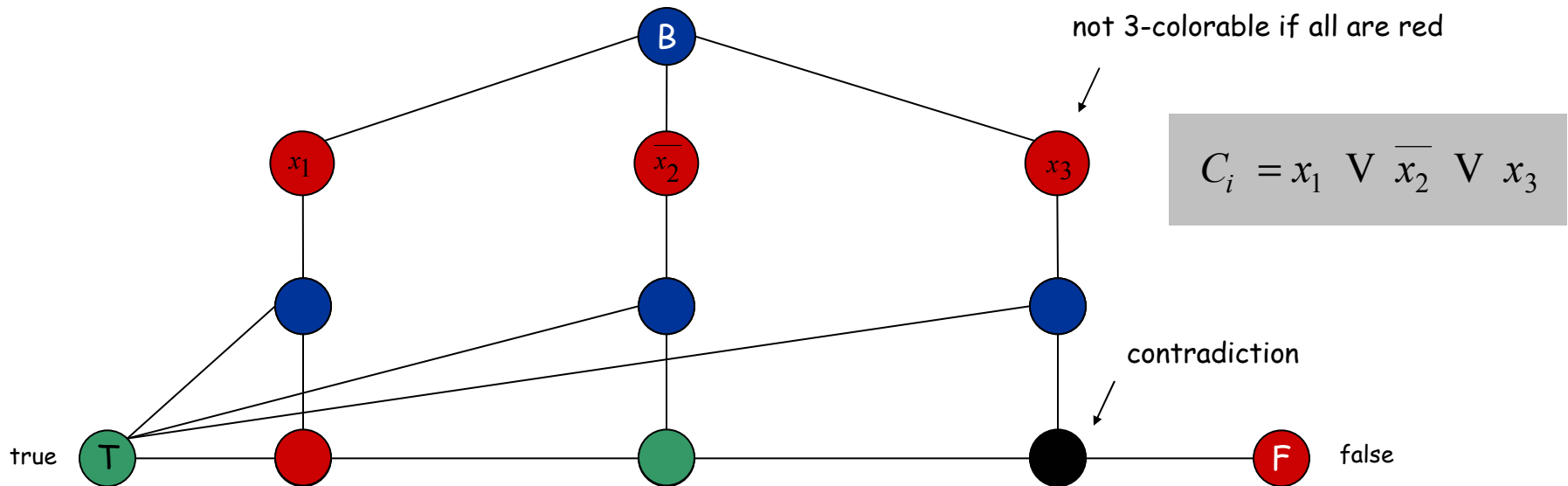


3-Colorability

Claim. Graph is 3-colorable iff Φ is satisfiable.

Pf. \Rightarrow Suppose graph is 3-colorable.

- Consider assignment that sets all T literals to true.
- (ii) ensures each literal is T or F.
- (iii) ensures a literal and its negation are opposites.
- (iv) ensures at least one literal in each clause is T.

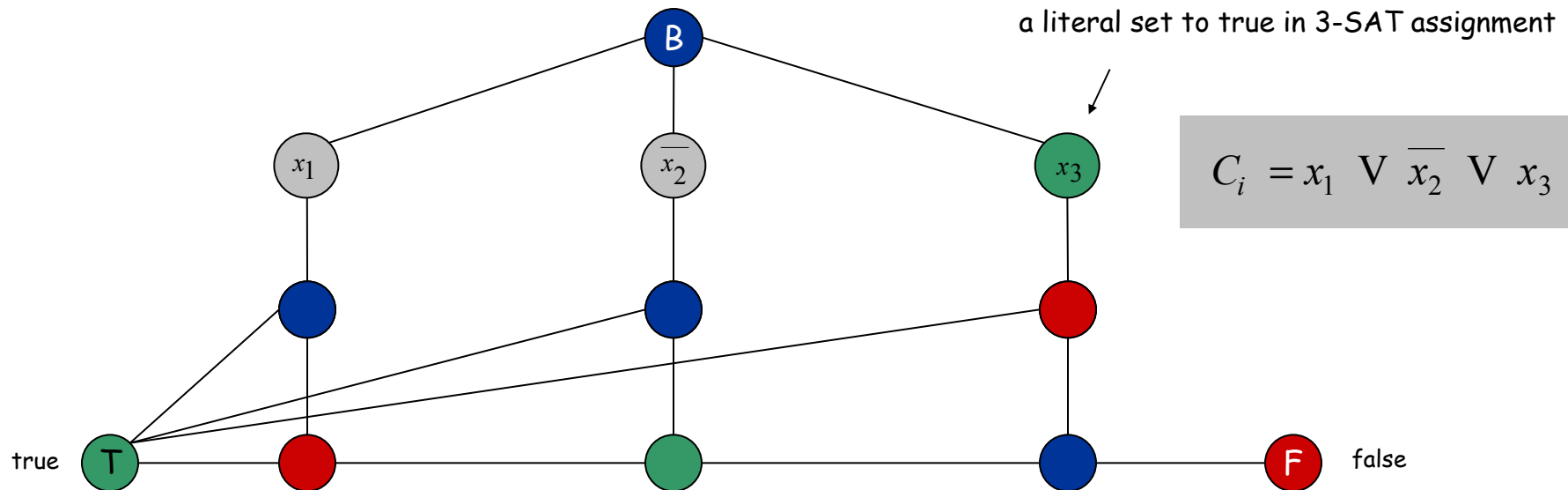


3-Colorability

Claim. Graph is 3-colorable iff Φ is satisfiable.

Pf. \Leftarrow Suppose 3-SAT formula Φ is satisfiable.

- Color all true literals T.
- Color node below green node F, and node below that B.
- Color remaining middle row nodes B.
- Color remaining bottom nodes T or F as forced. ▪



8.8 Numerical Problems

Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3-COLOR, 3D-MATCHING.
- **Numerical problems:** SUBSET-SUM, KNAPSACK.

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.

$$\begin{aligned}
 C_1 &= \bar{x} \vee y \vee z \\
 C_2 &= x \vee \bar{y} \vee z \\
 C_3 &= \bar{x} \vee \bar{y} \vee \bar{z}
 \end{aligned}$$

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	100,110
$\neg x$	1	0	0	1	0	1	100,001
y	0	1	0	1	0	0	10,000
$\neg y$	0	1	0	0	1	1	10,111
z	0	0	1	1	1	0	1,010
$\neg z$	0	0	1	0	0	1	1,101
	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

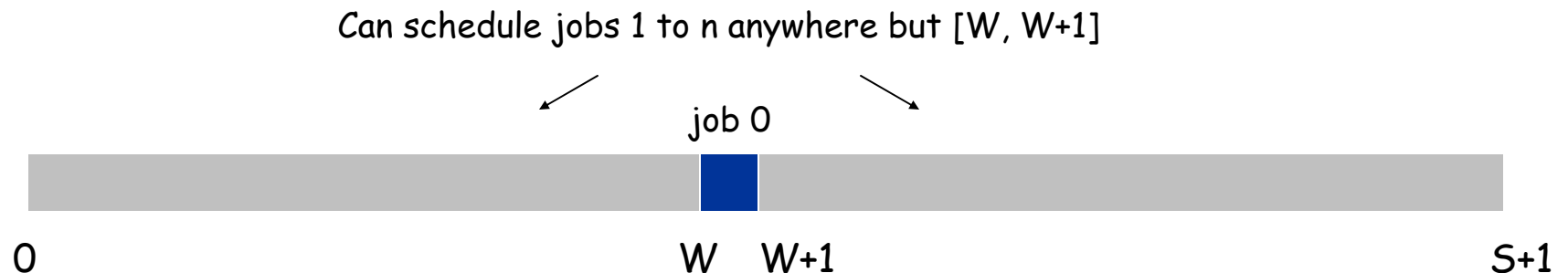
Scheduling With Release Times

SCHEDULE-RELEASE-TIMES. Given a set of n jobs with processing time t_i , release time r_i , and deadline d_i , is it possible to schedule all jobs on a single machine such that job i is processed with a contiguous slot of t_i time units in the interval $[r_i, d_i]$?

Claim. $\text{SUBSET-SUM} \leq_p \text{SCHEDULE-RELEASE-TIMES}$.

Pf. Given an instance of SUBSET-SUM w_1, \dots, w_n , and target W ,

- Create n jobs with processing time $t_i = w_i$, release time $r_i = 0$, and no deadline ($d_i = 1 + \sum_j w_j$).
- Create job 0 with $t_0 = 1$, release time $r_0 = W$, and deadline $d_0 = W+1$.

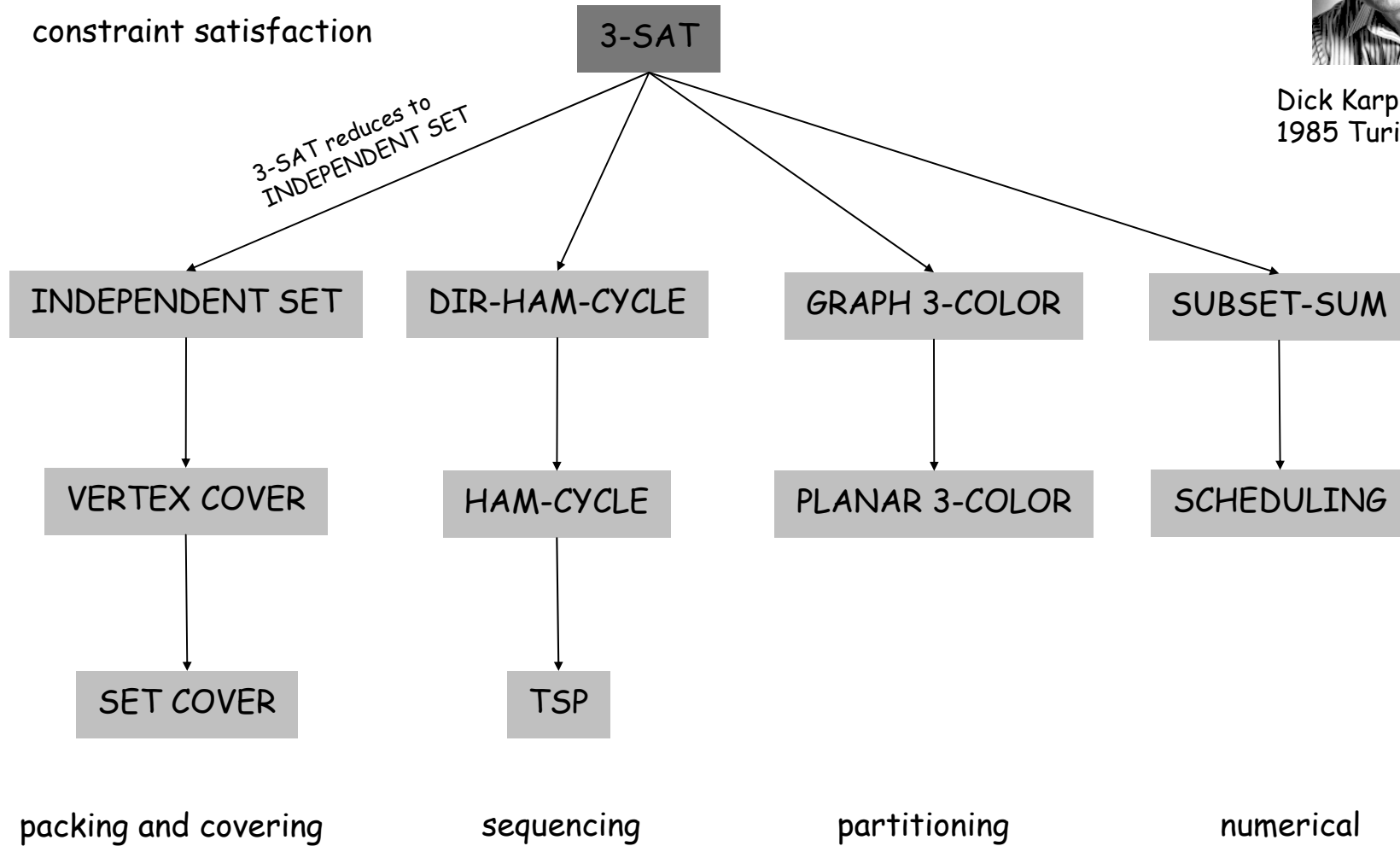


8.10 A Partial Taxonomy of Hard Problems

Polynomial-Time Reductions



Dick Karp (1972)
1985 Turing Award



Extra Slides

Subset Sum (proof from book)

Construction. Let $X \cup Y \cup Z$ be an instance of 3D-MATCHING with triplet set T . Let $n = |X| = |Y| = |Z|$ and $m = |T|$.

- Let $X = \{x_1, x_2, x_3, x_4\}$, $Y = \{y_1, y_2, y_3, y_4\}$, $Z = \{z_1, z_2, z_3, z_4\}$
- For each triplet $t = (x_i, y_j, z_k) \in T$, create an integer w_t with $3n$ digits that has a 1 in positions i , $n+j$, and $2n+k$.

\uparrow
 use base $m+1$

Claim. 3D-matching iff some subset sums to $W = 111, \dots, 111$.

Triplet t_i			x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4	z_1	z_2	z_3	z_4	w_i
x_1	y_2	z_3	1	0	0	0	0	1	0	0	0	0	1	0	100,001,000,010
x_2	y_4	z_2	0	1	0	0	0	0	0	1	0	1	0	0	10,000,010,100
x_1	y_1	z_1	1	0	0	0	1	0	0	0	1	0	0	0	100,010,001,000
x_2	y_2	z_4	0	1	0	0	0	1	0	0	0	0	0	1	10,001,000,001
x_4	y_3	z_4	0	0	0	1	0	0	1	0	0	0	0	1	100,100,001
x_3	y_1	z_2	0	0	1	0	1	0	0	0	0	1	0	0	1,010,000,100
x_3	y_1	z_3	0	0	1	0	1	0	0	0	0	0	1	0	1,010,000,010
x_3	y_1	z_1	0	0	1	0	1	0	0	0	1	0	0	0	1,010,001,000
x_4	y_4	z_4	0	0	0	1	0	0	0	1	0	0	0	1	100,010,001
															111,111,111,111