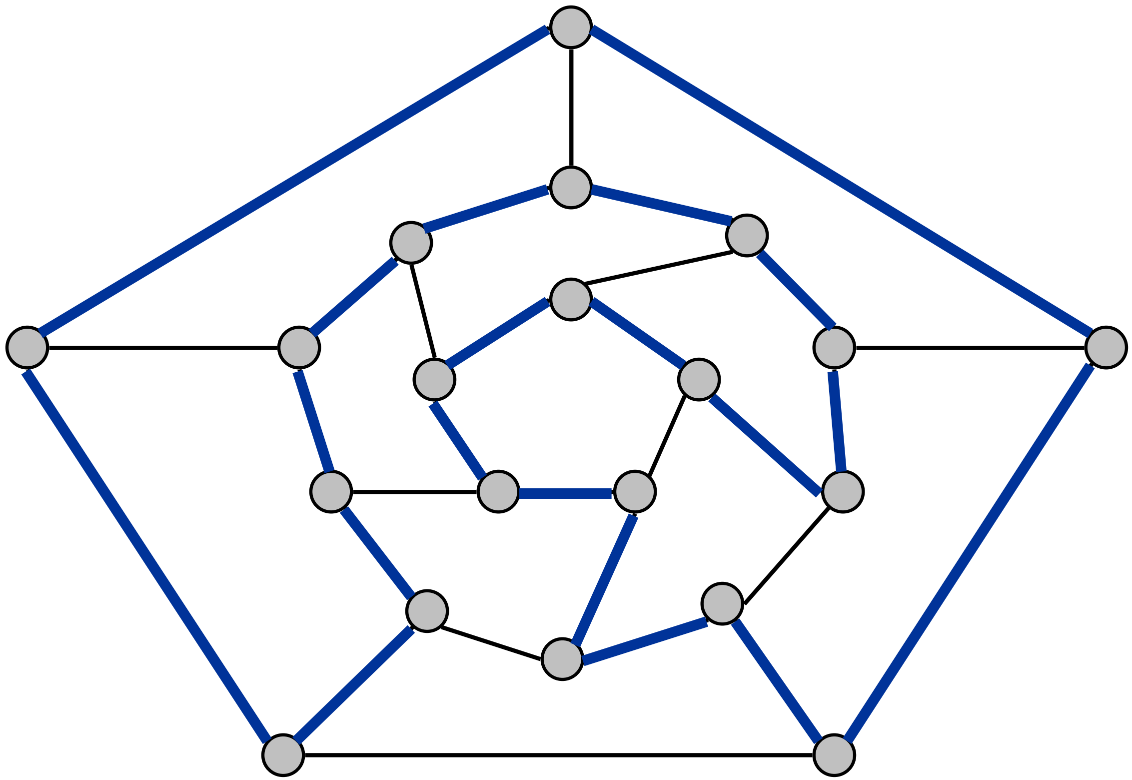
8.5 Sequencing Problems

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

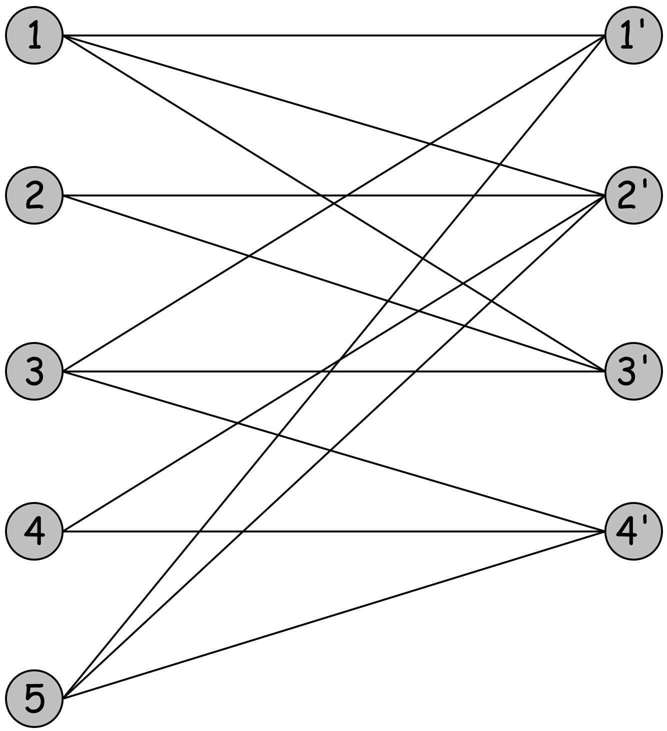
HAM-CYCLE: given an undirected graph $G = (V, E)$, does there exist a simple cycle Γ that contains every node in V.



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.



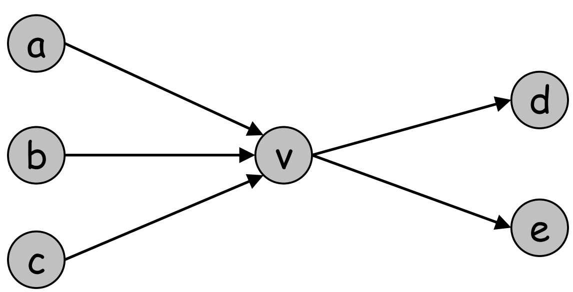
NO: bipartite graph with odd number of nodes.

Directed Hamiltonian Cycle

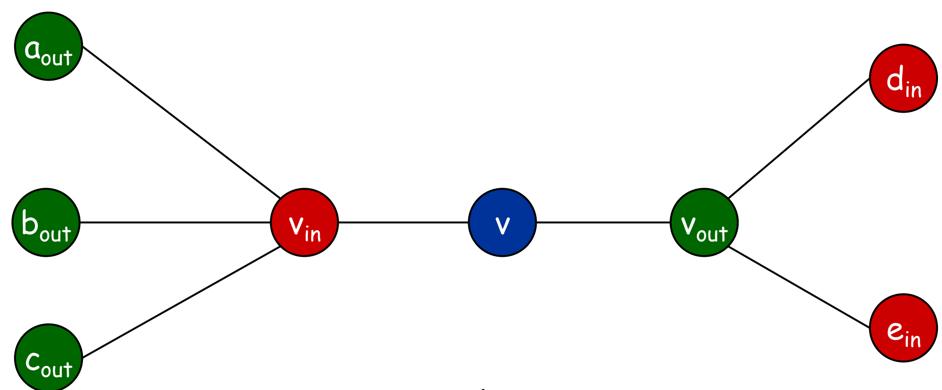
DIR-HAM-CYCLE: given a digraph $G = (V, E)$, does there exists a simple directed cycle Γ that contains every node in V?

Claim. DIR-HAM-CYCLE $\leq_P$ HAM-CYCLE.

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



G



G'

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-SAT $\leq_P$ 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.

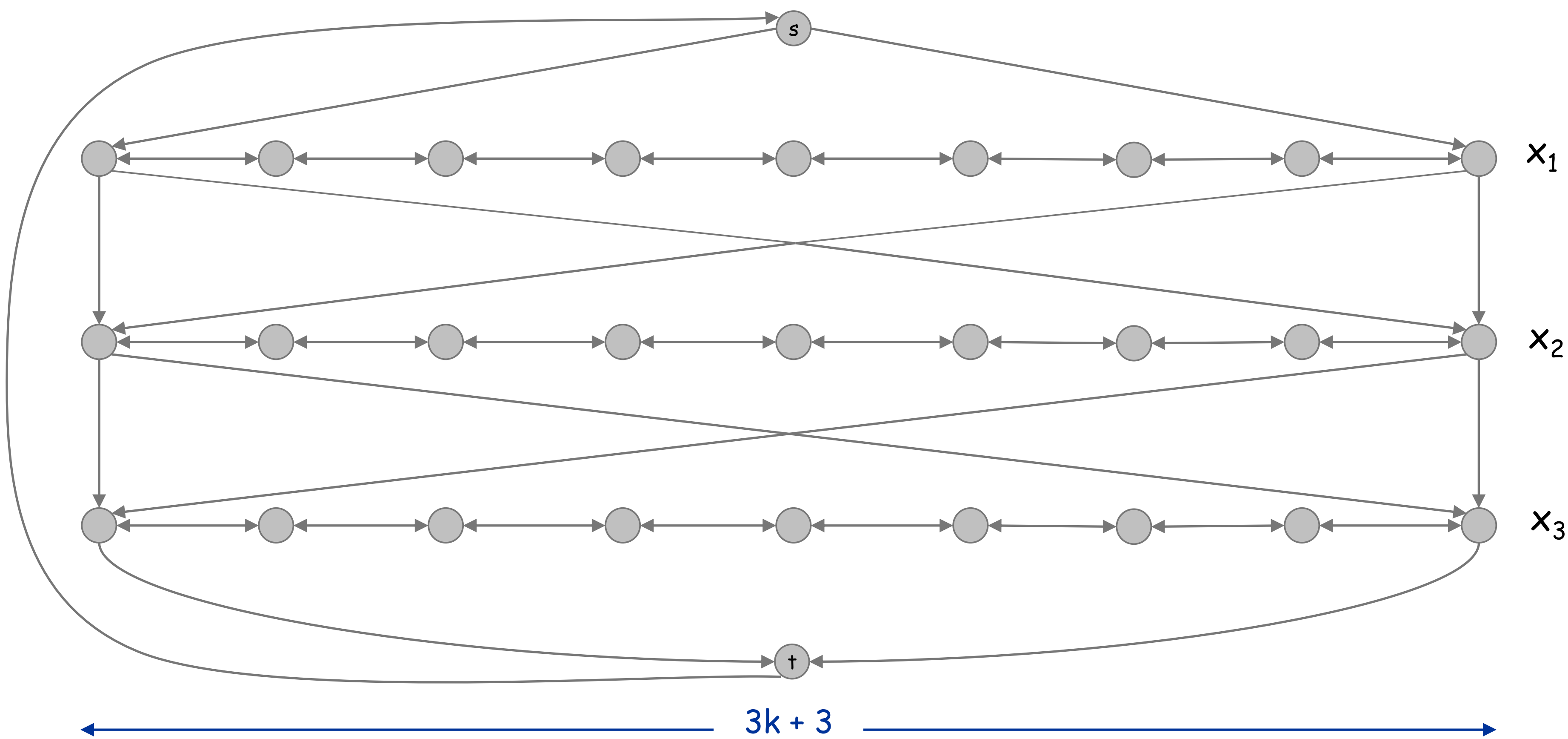
Construction. First, create graph that has 2^n Hamiltonian cycles which correspond in a natural way to 2^n possible truth assignments.

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.

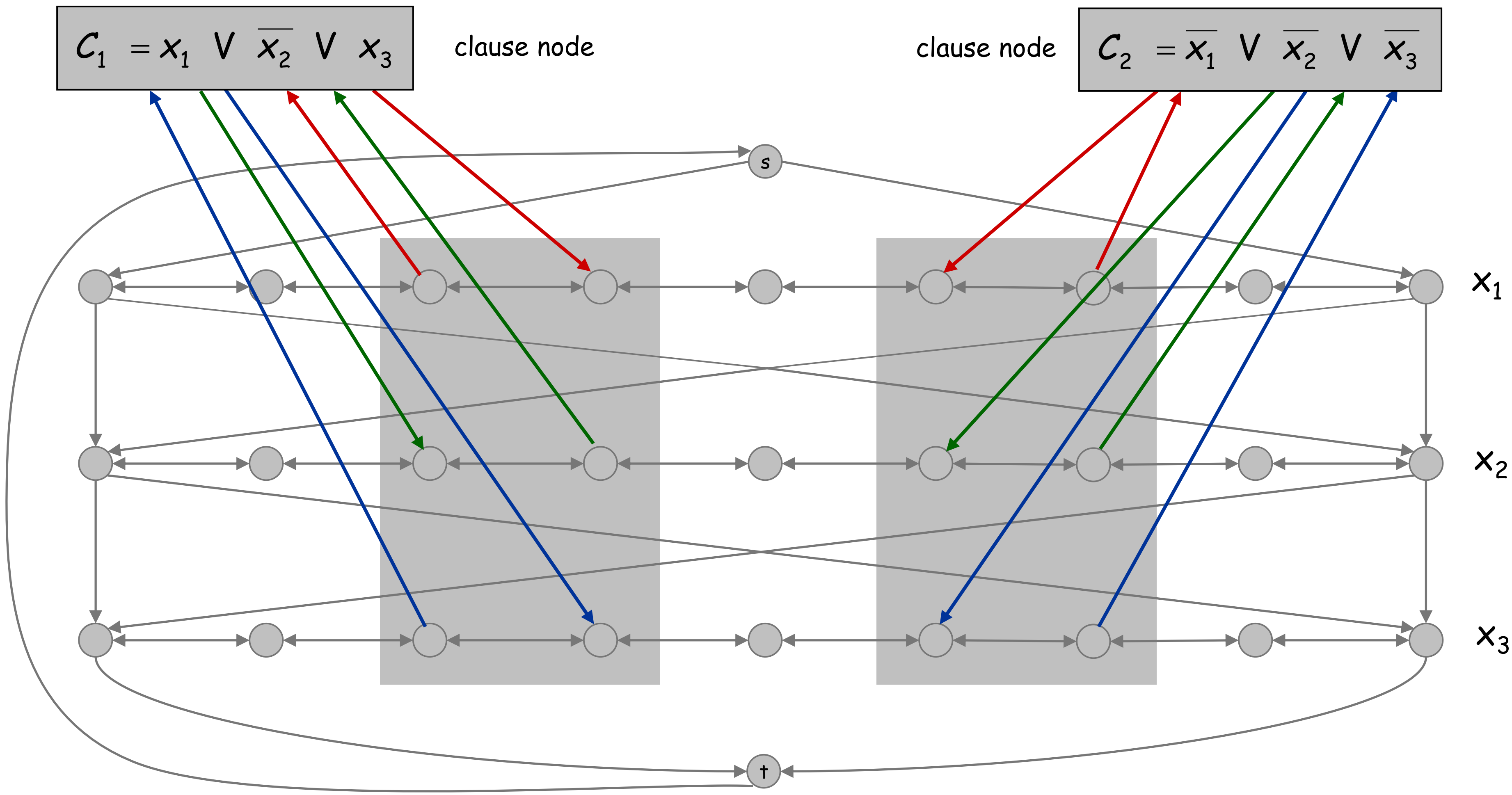
Intuition: traverse path i from left to right $\Leftrightarrow$ set variable $x_i = 1$.



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 $G - \{ C_1, C_2, \ldots, C_k \}$.

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 exists a simple path of length at most k edges?

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

Claim. 3-SAT $\leq_P$ LONGEST-PATH.

Pf 1. Redo proof for DIR-HAM-CYCLE, ignoring back-edge from t to s.
Pf 2. Show HAM-CYCLE $\leq_P$ 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难捉摸的:
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.

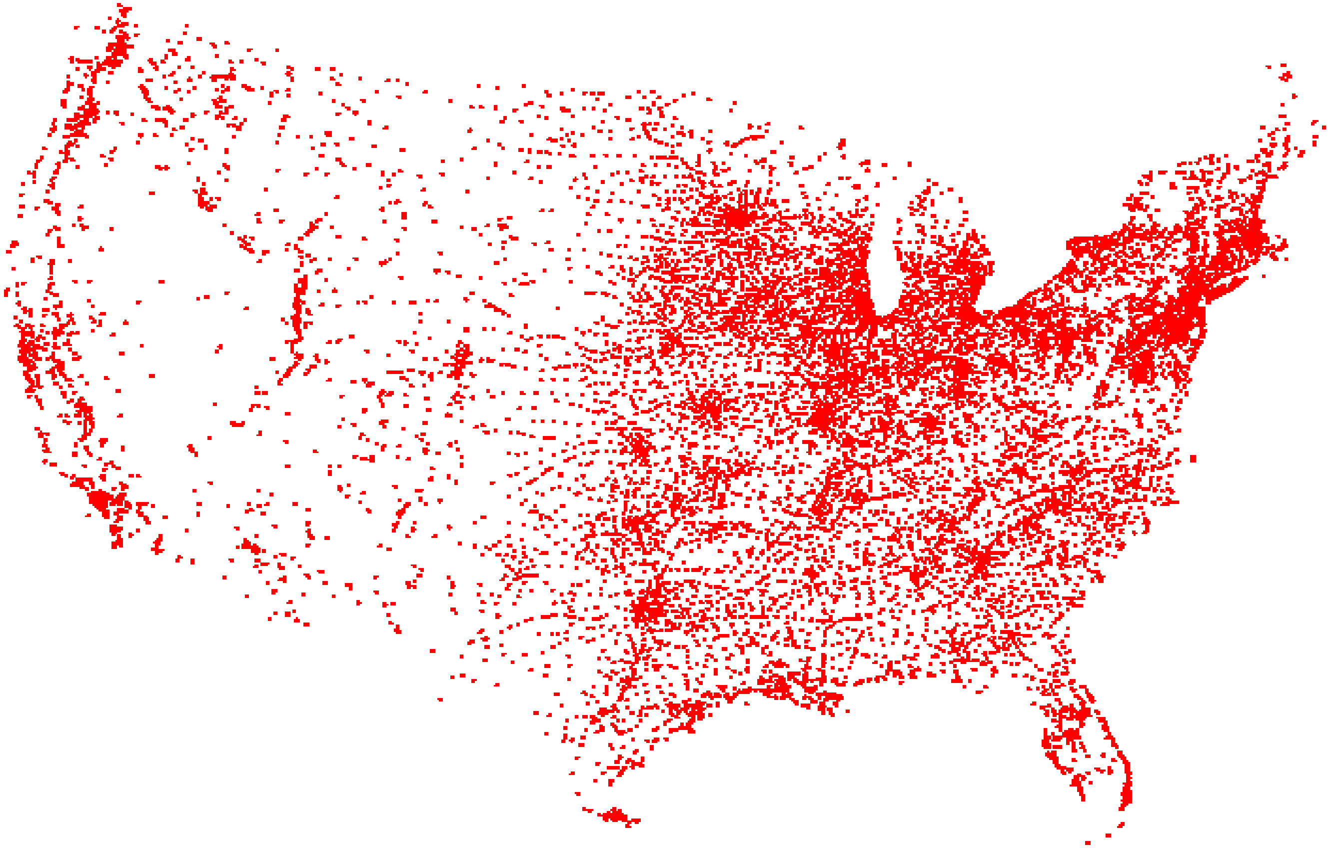
Garey, Johnson, Karp and other men (and women)
Tried to make it order N log N.
Am I a mad fool
If I spend my life in grad school,
Forever following the longest path?

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

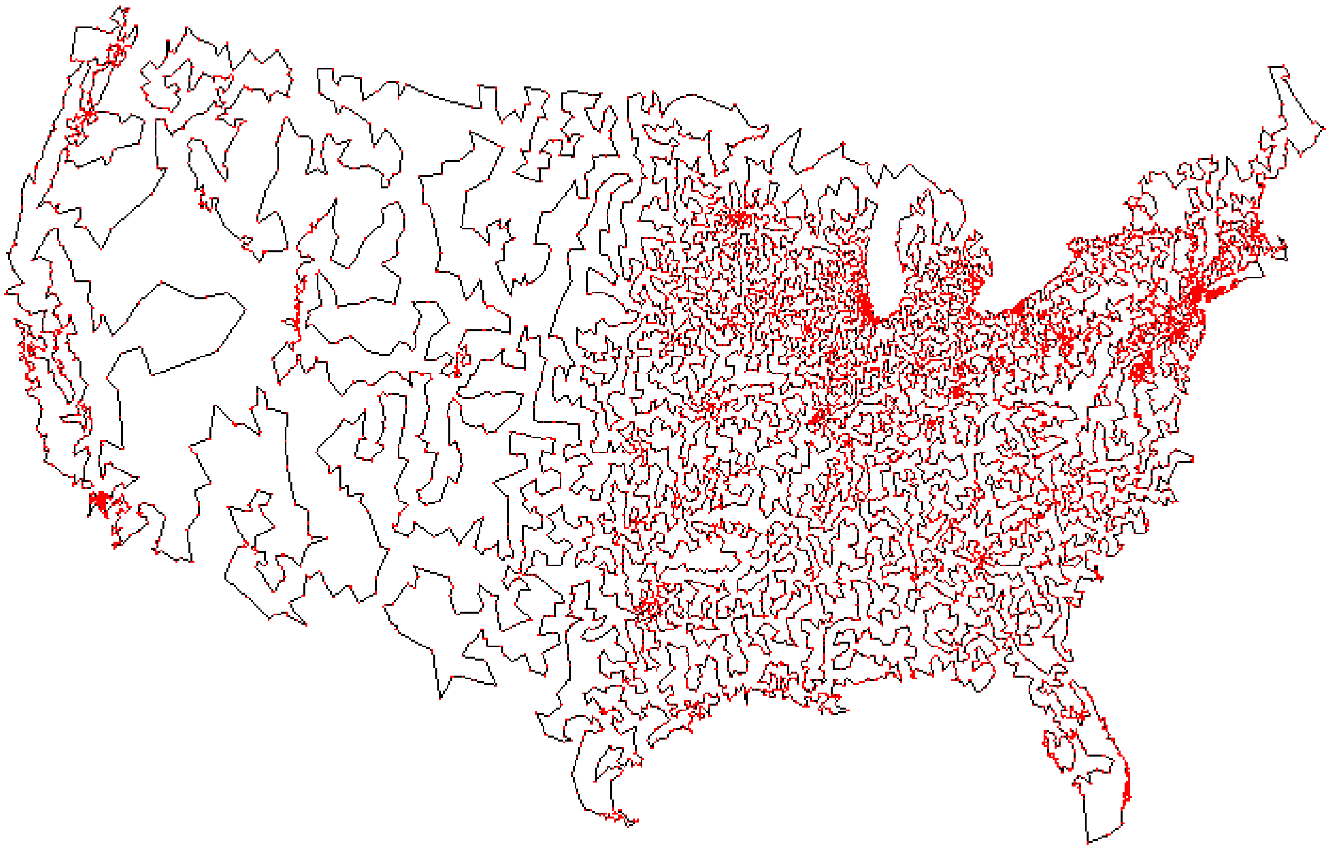
TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length $\leq$ D?



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

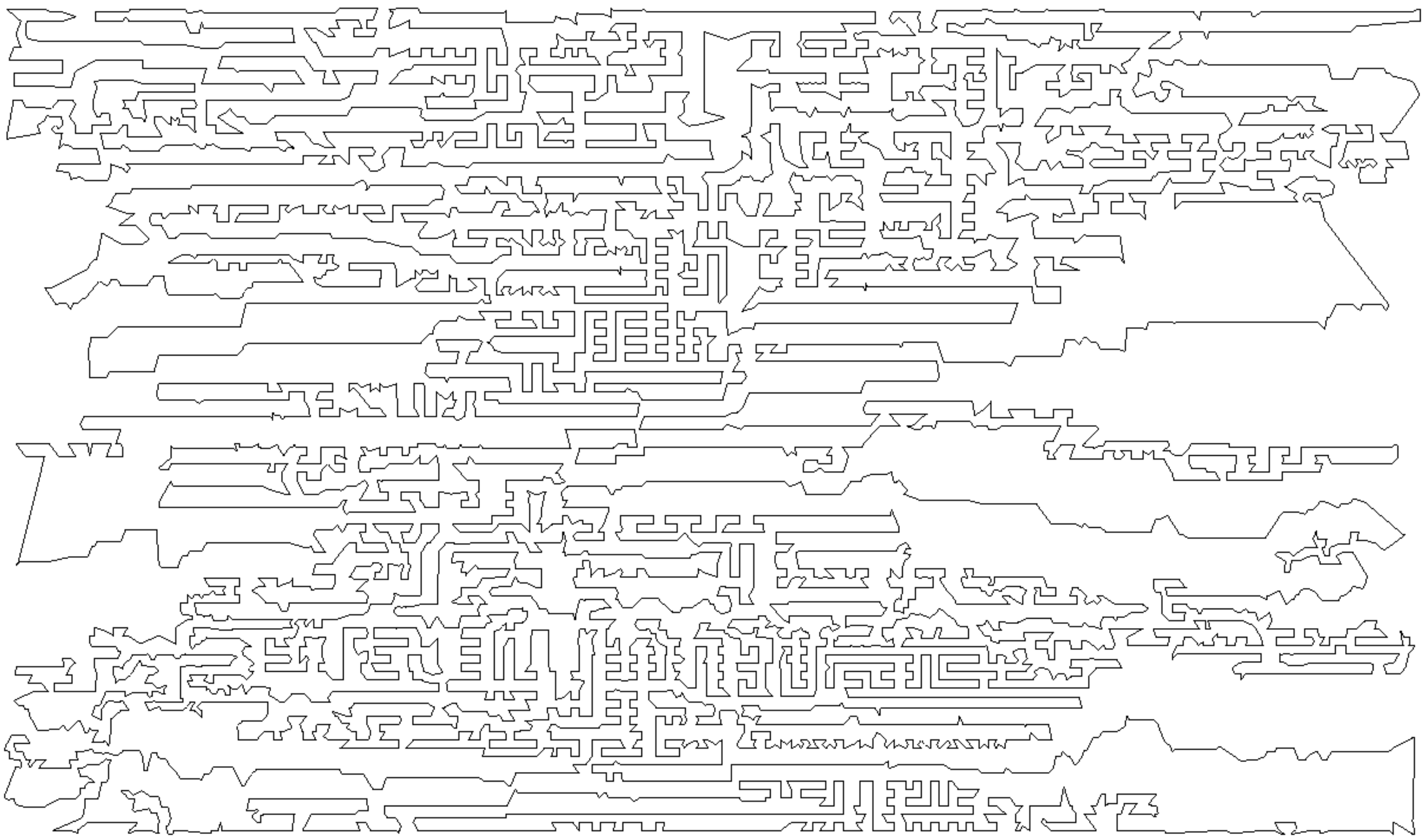
Traveling Salesperson Problem

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 exists a simple cycle that contains every node in V?

Claim. HAM-CYCLE $\leq_P$ 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-SAT $\leq_P$ INDEPENDENT-COVER.

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

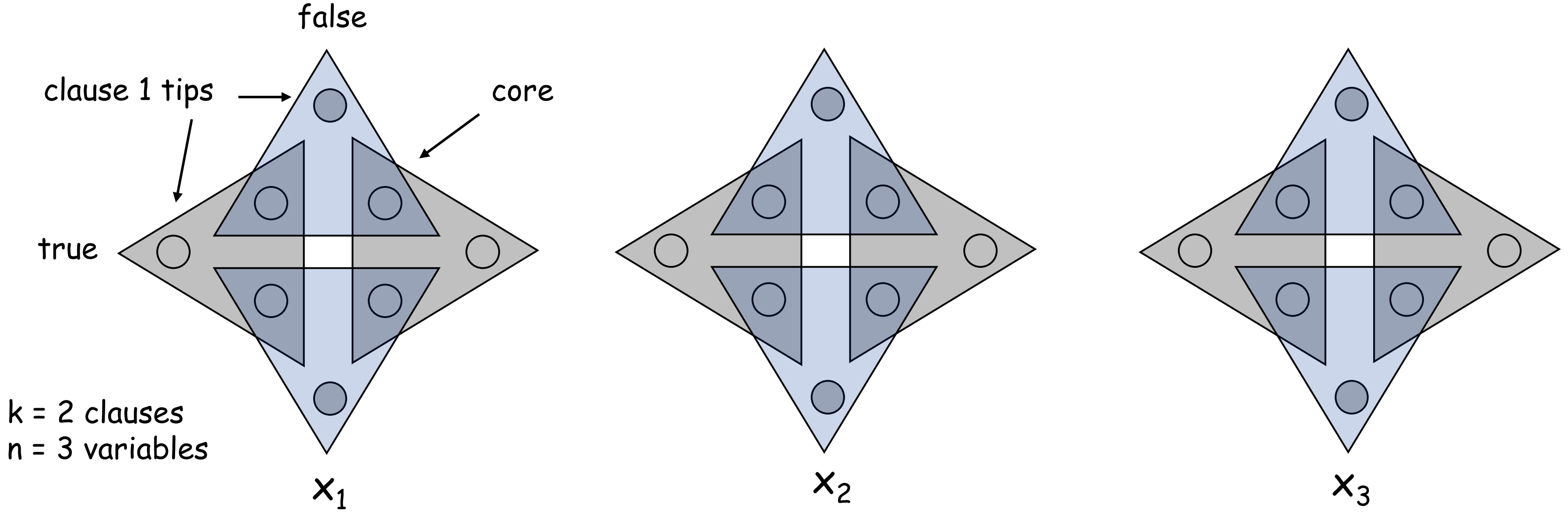
3-Dimensional Matching

Construction. (part 1)

number of clauses

- 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.

set x_i = true   set x_i = false



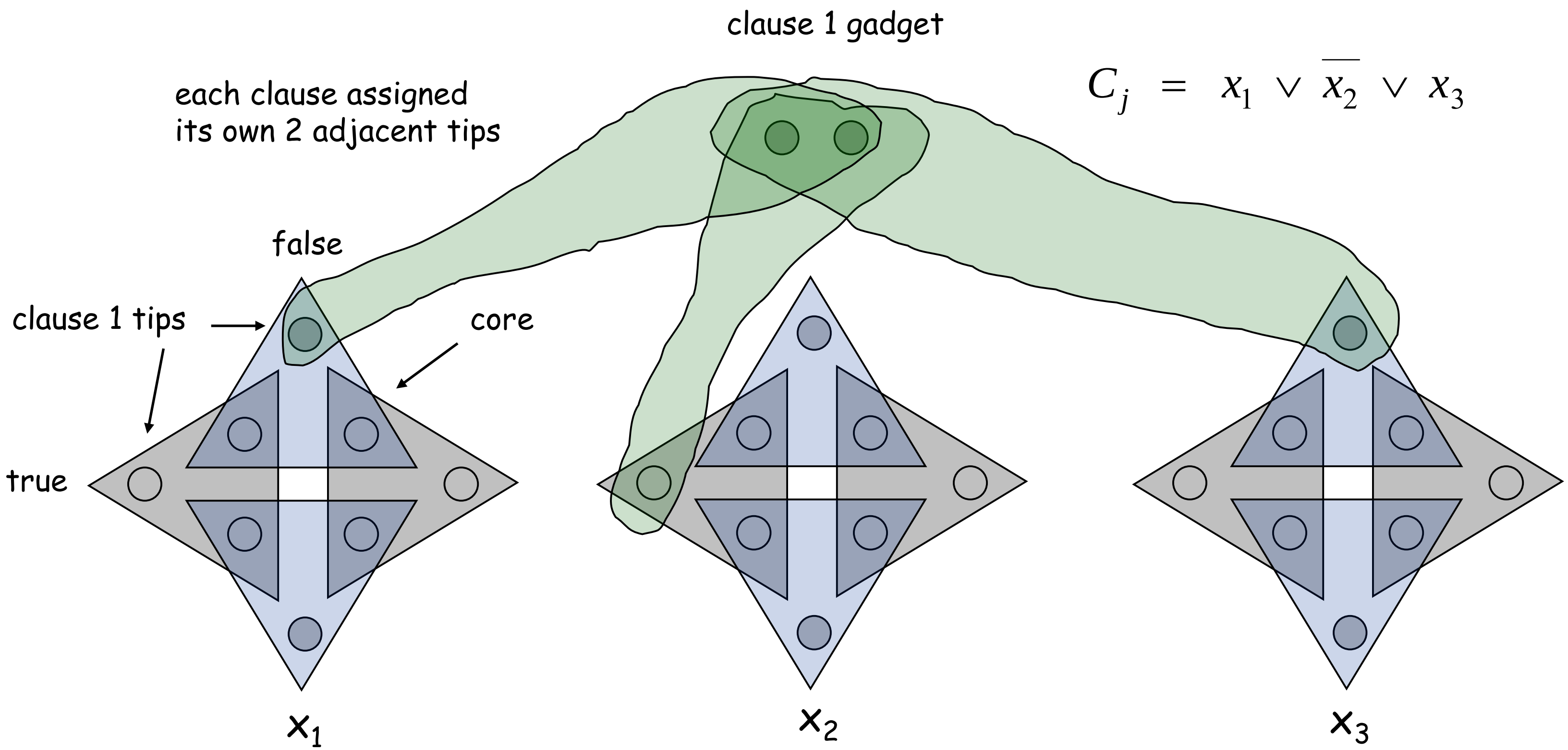
3-Dimensional Matching

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.

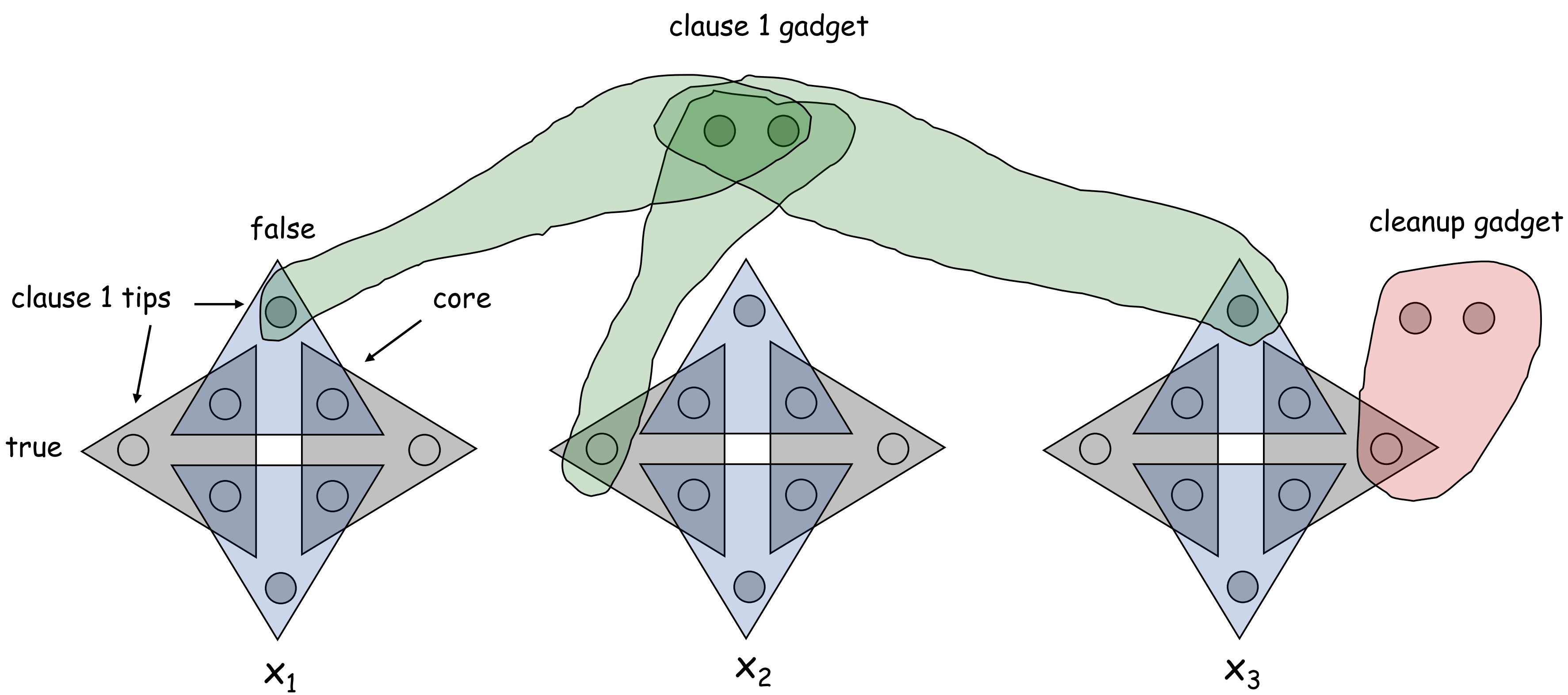
Ensures any 3D-matching uses either (i) grey core of x_1 or (ii) blue core of x_2 or (iii) grey core of x_3.



3-Dimensional Matching

Construction. (part 3)

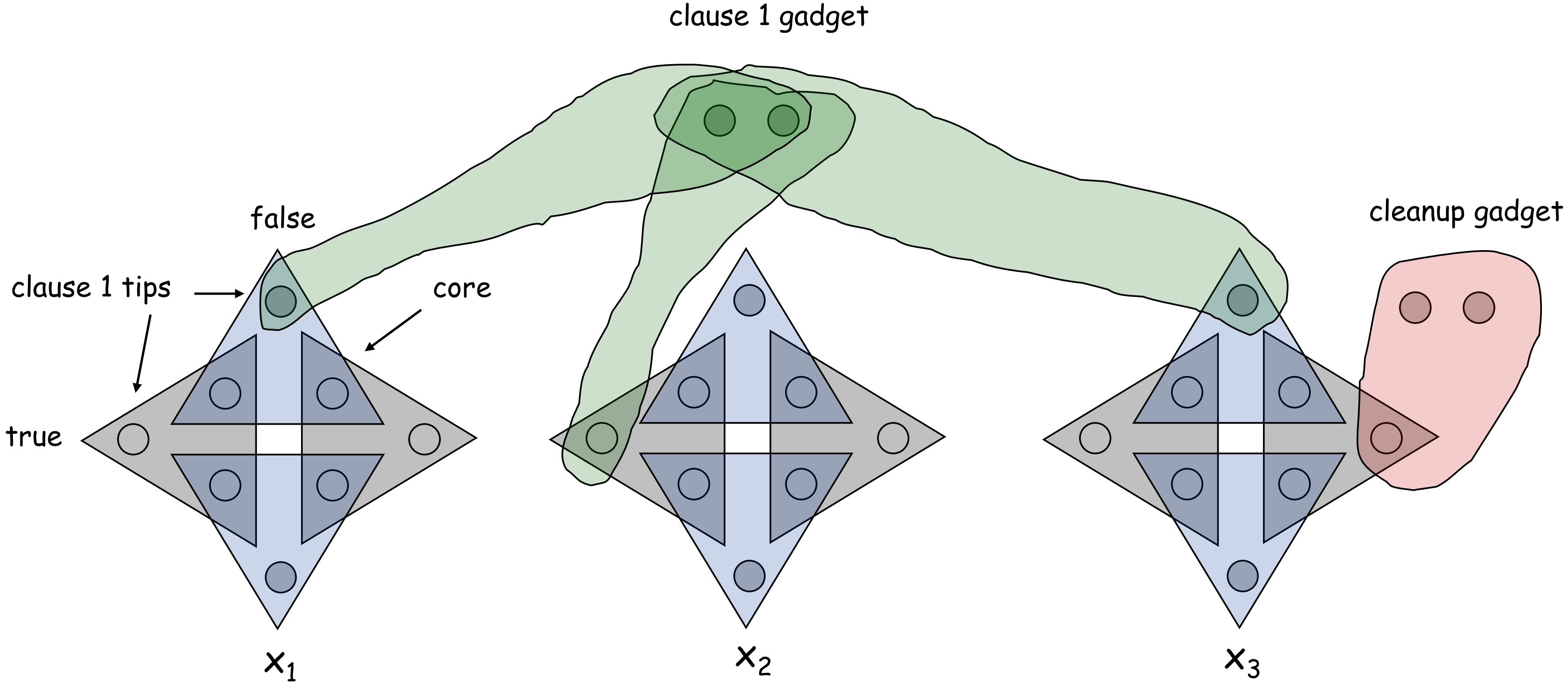
For each tip, add a cleanup gadget.



3-Dimensional Matching

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

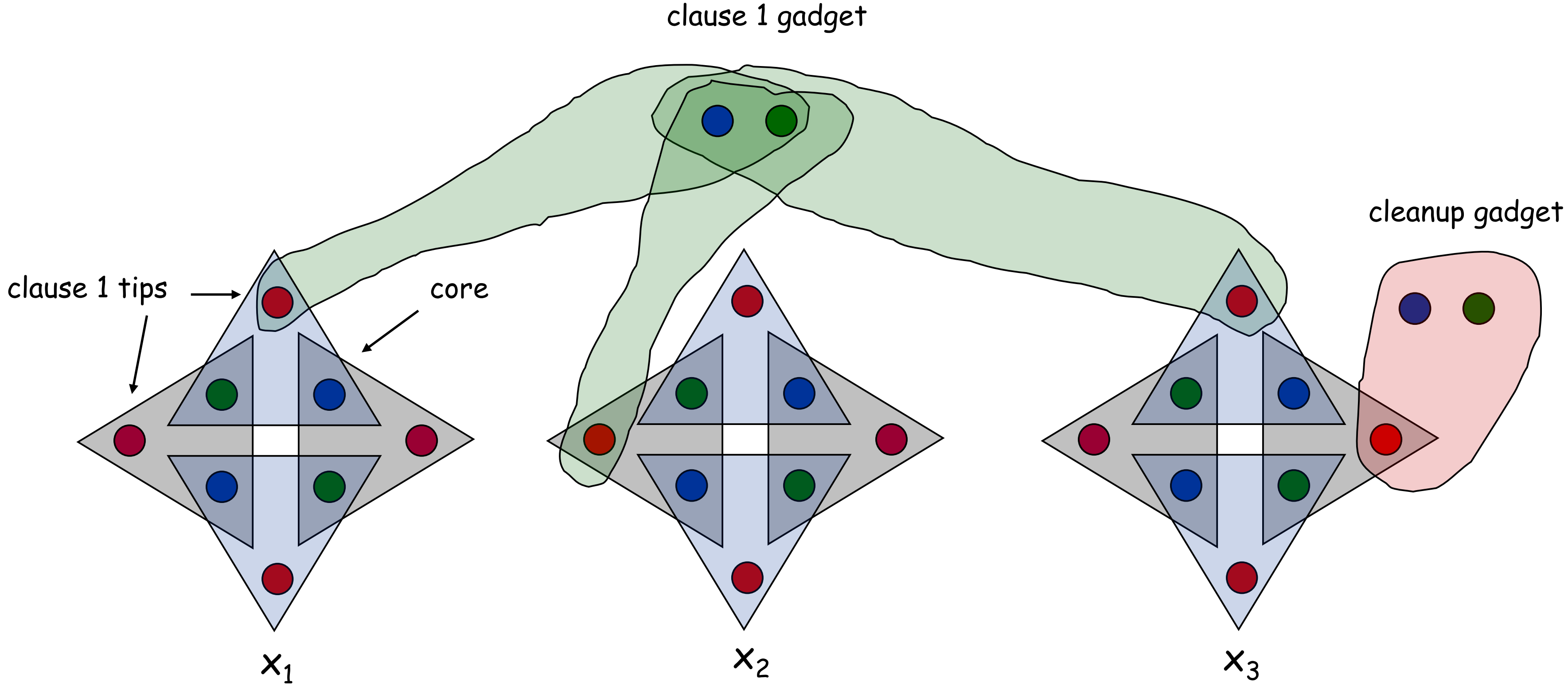
Detail. What are X, Y, and Z? Does each triple contain one element from each of X, Y, Z?



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?



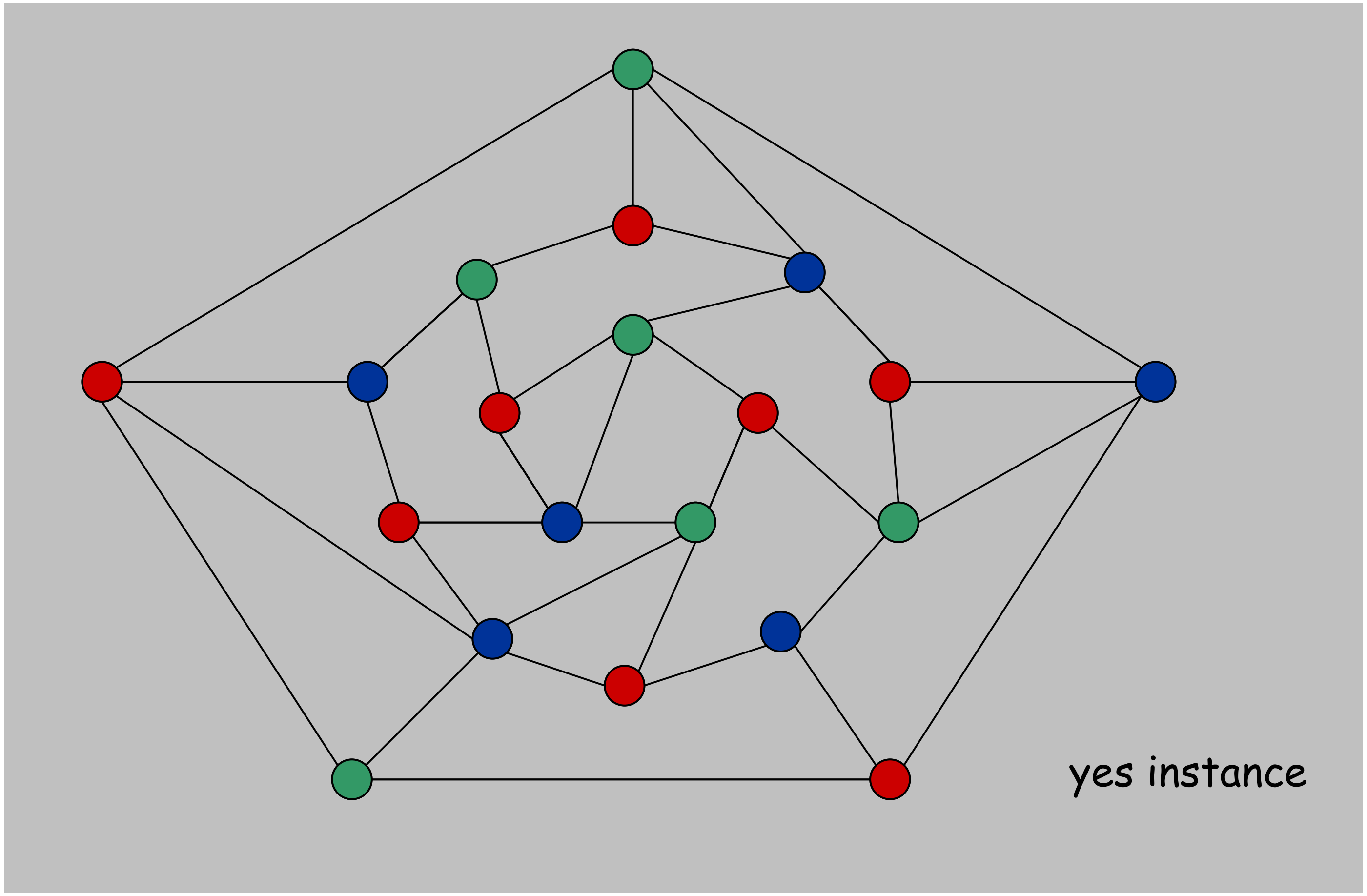
8.7 Graph Coloring

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 exists 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-COLOR $\leq_P$ k-REGISTER-ALLOCATION for any constant $k \geq 3$.

3-Colorability

Claim. 3-SAT $\leq_P$ 3-COLOR.

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

Construction.

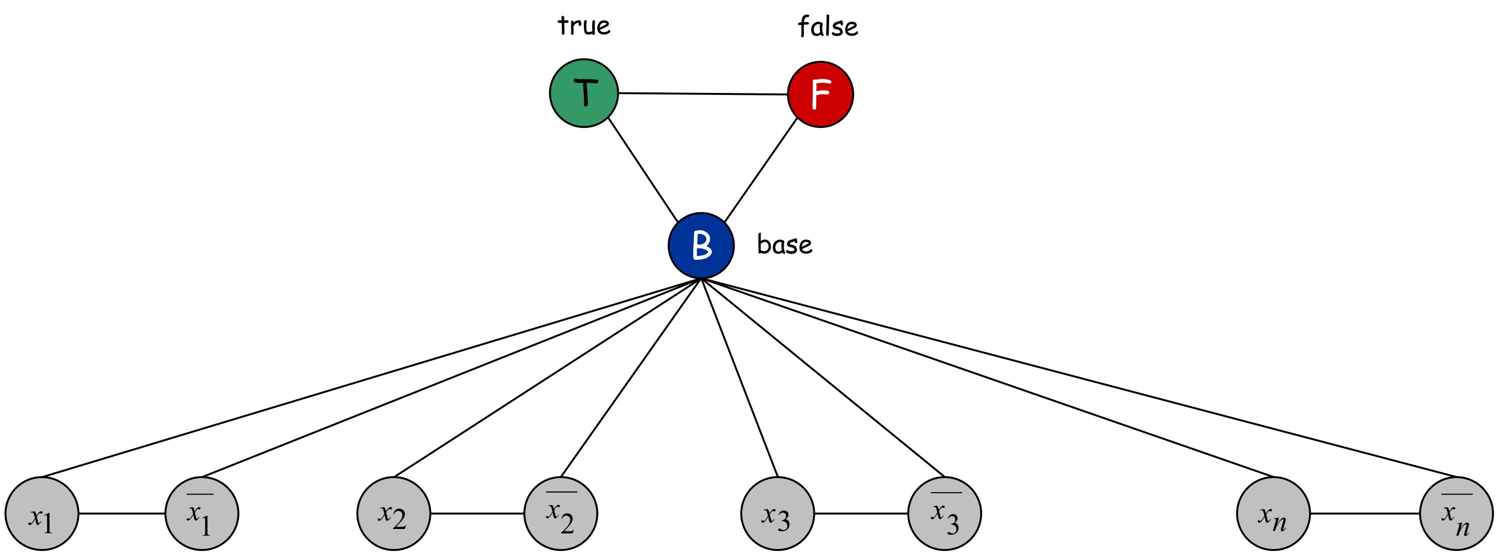
i. For each literal, create a node.
ii. Create 3 new nodes T, F, B; connect them in a triangle, and connect each literal to B.
iii. Connect each literal to its negation.
iv. For each clause, add gadget of 6 nodes and 13 edges.

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.

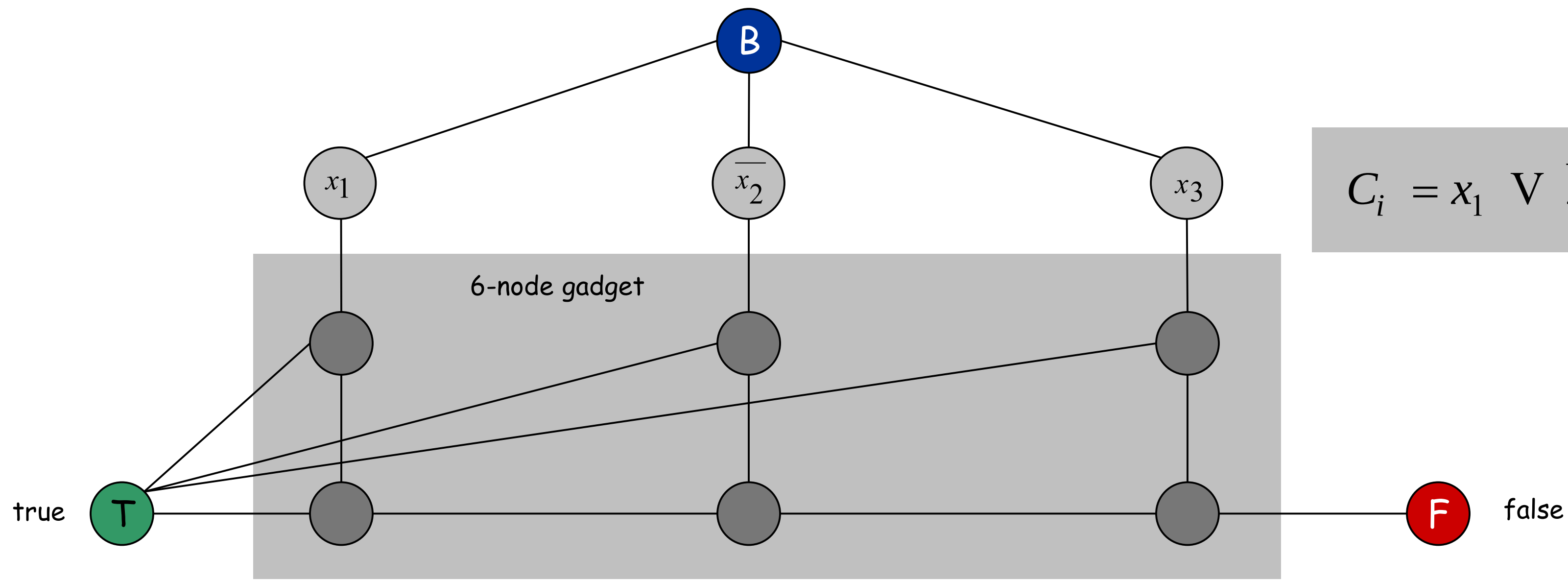
Pf. $\Rightarrow$ Suppose graph is 3-colorable.

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.



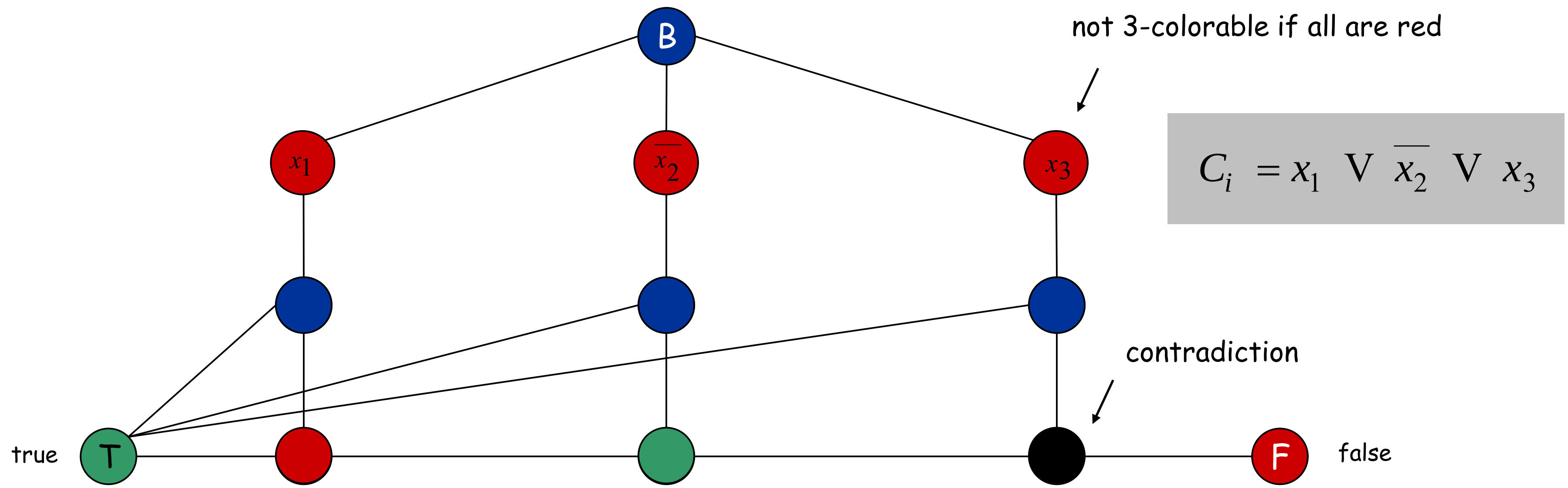
$$C_i = x_1 \lor \overline{x_2} \lor x_3$$

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.

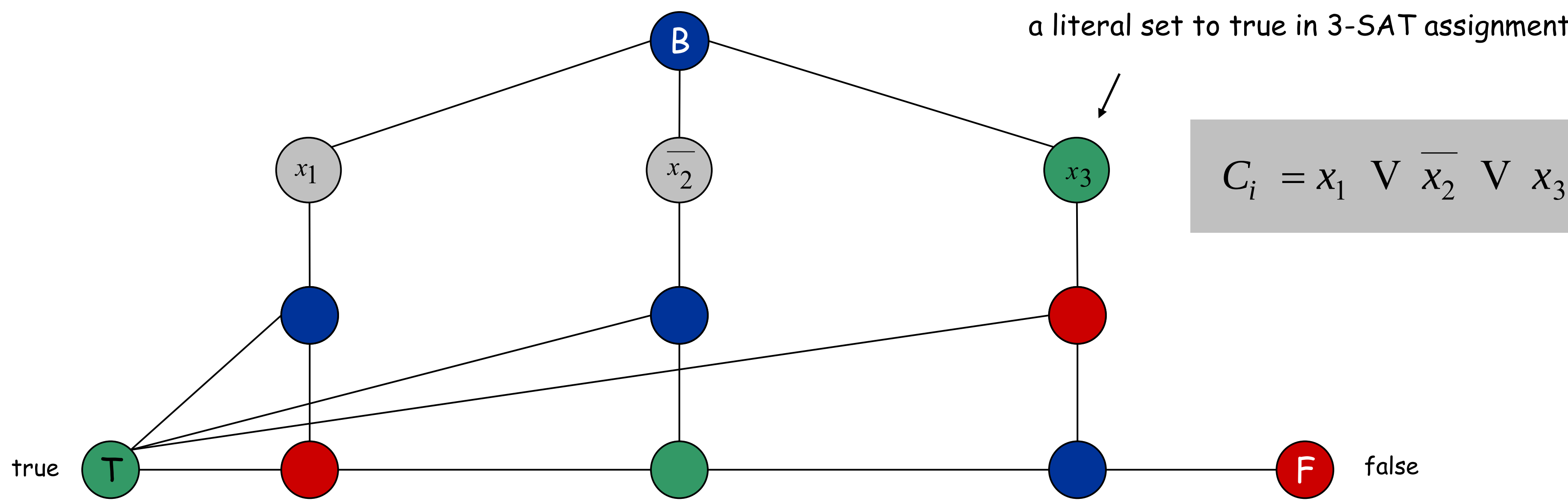


3-Colorability

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. ▪



$$C_i = x_1 \vee \overline{x_2} \vee x_3$$

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 $w_1, \ldots, w_n$ 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-SAT $\leq_P$ 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.

$$C_1 = \bar{x} \vee y \vee z$$
$$C_2 = x \vee \bar{y} \vee z$$
$$C_3 = \bar{x} \vee \bar{y} \vee \bar{z}$$

	x	y	z	C_1	C_2	C_3	
x	1	0	0	0	1	0	100,110
¬ x	1	0	0	1	0	1	100,101
y	0	1	0	1	0	0	10,100
¬ y	0	1	0	0	1	1	10,011
z	0	0	1	1	1	0	1,110
¬ 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

dummies to get clause columns to sum to 4

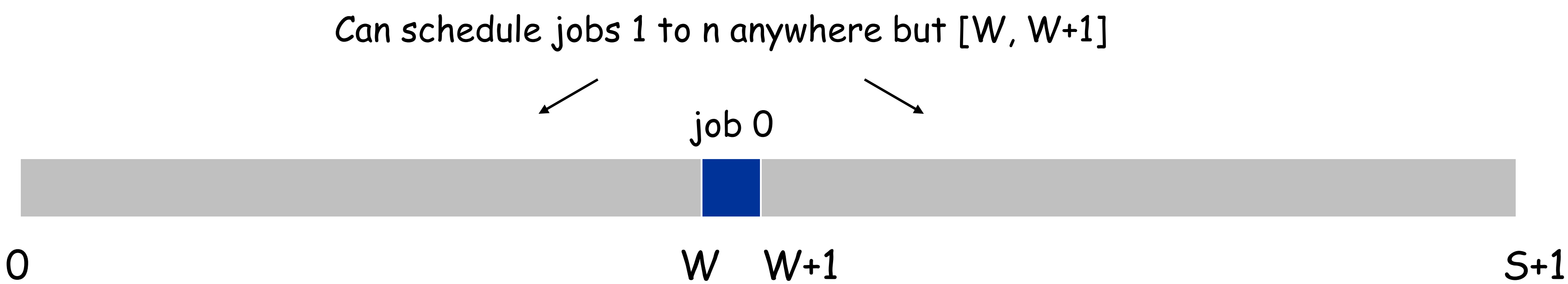
Scheduling With Release Times

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. SUBSET-SUM $\le_P$ SCHEDULE-RELEASE-TIMES.

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

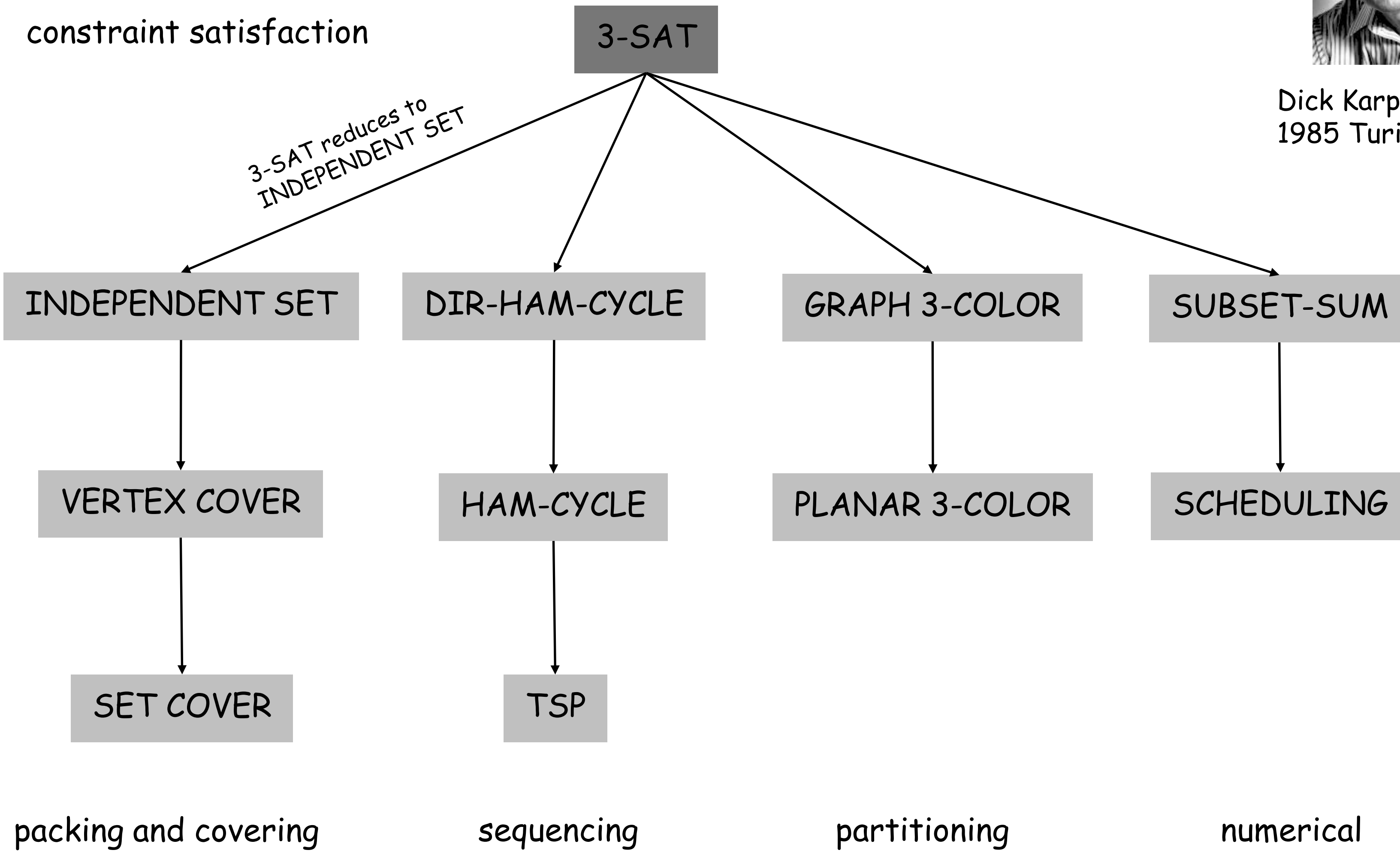
- Create n jobs with processing time $t_i = w_i$, release time $r_i = 0$, and no deadline ($d_i = 1 + \Sigma_j w_j$).
- Create job 0 with $t_0 = 1$, release time $r_0 = W$, and deadline $d_0 = W+1$.



8.10 A Partial Taxonomy of Hard Problems

Polynomial-Time Reductions

Dick Karp (1972)
1985 Turing Award



Homeworks

Homework: 8.3, 8.5, 8.7, 8.16,8.39

Extra Slides

Subset Sum (proof from book)

Construction. Let $X \cup Y \cup Z$ be a 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.

use base m+1

Claim. 3D-matching iff some subset sums to W = 111,…, 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

Partition

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

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

$\frac{1}{2} \Sigma_i v_i$

Claim. SUBSET-SUM $\leq_P$ PARTITION.

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

Create instance of PARTITION with $m = n+2$ elements.

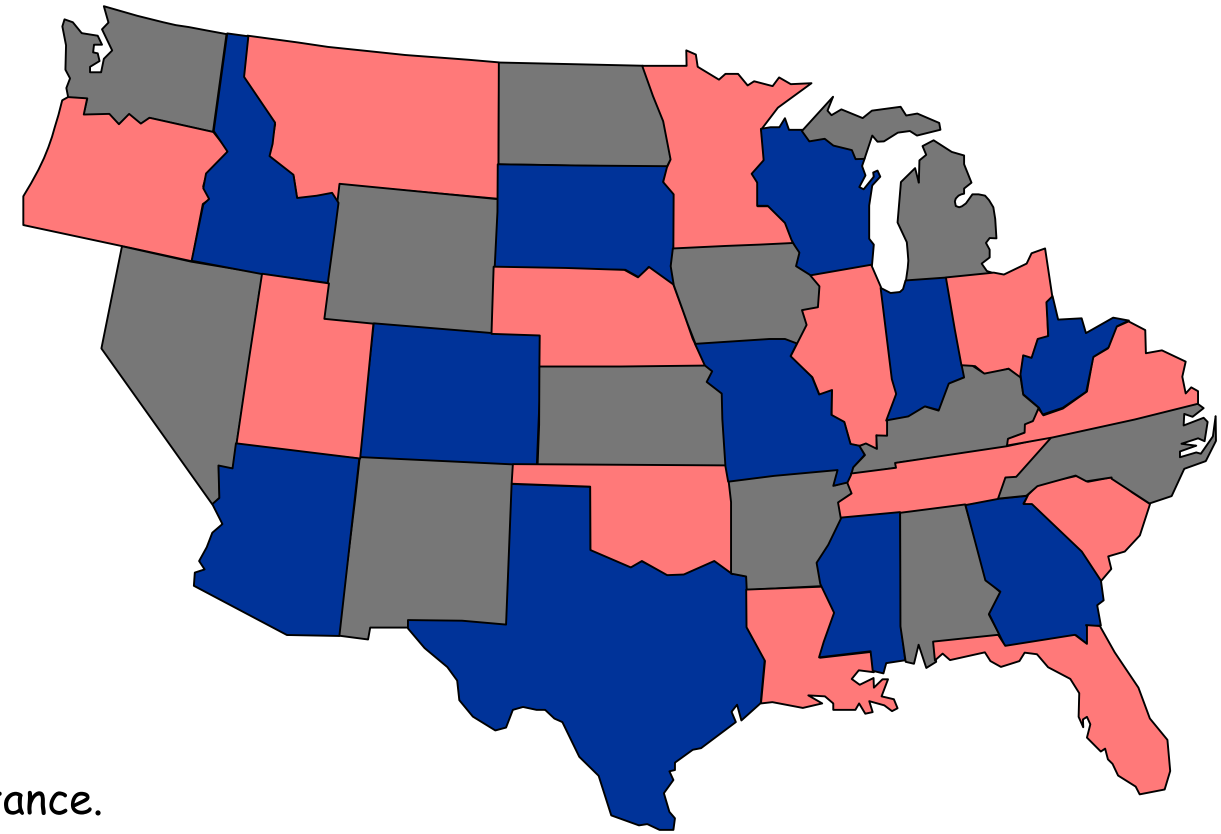
- $v_1 = w_1, v_2 = w_2, \ldots, v_n = w_n, \quad v_{n+1} = 2 \Sigma_i w_i - W, \quad v_{n+2} = \Sigma_i w_i + W$

There exists a subset that sums to W iff there exists a partition since two new elements cannot be in the same partition. ▪

Extra Slides: 4 Color Theorem

Planar 3-Colorability

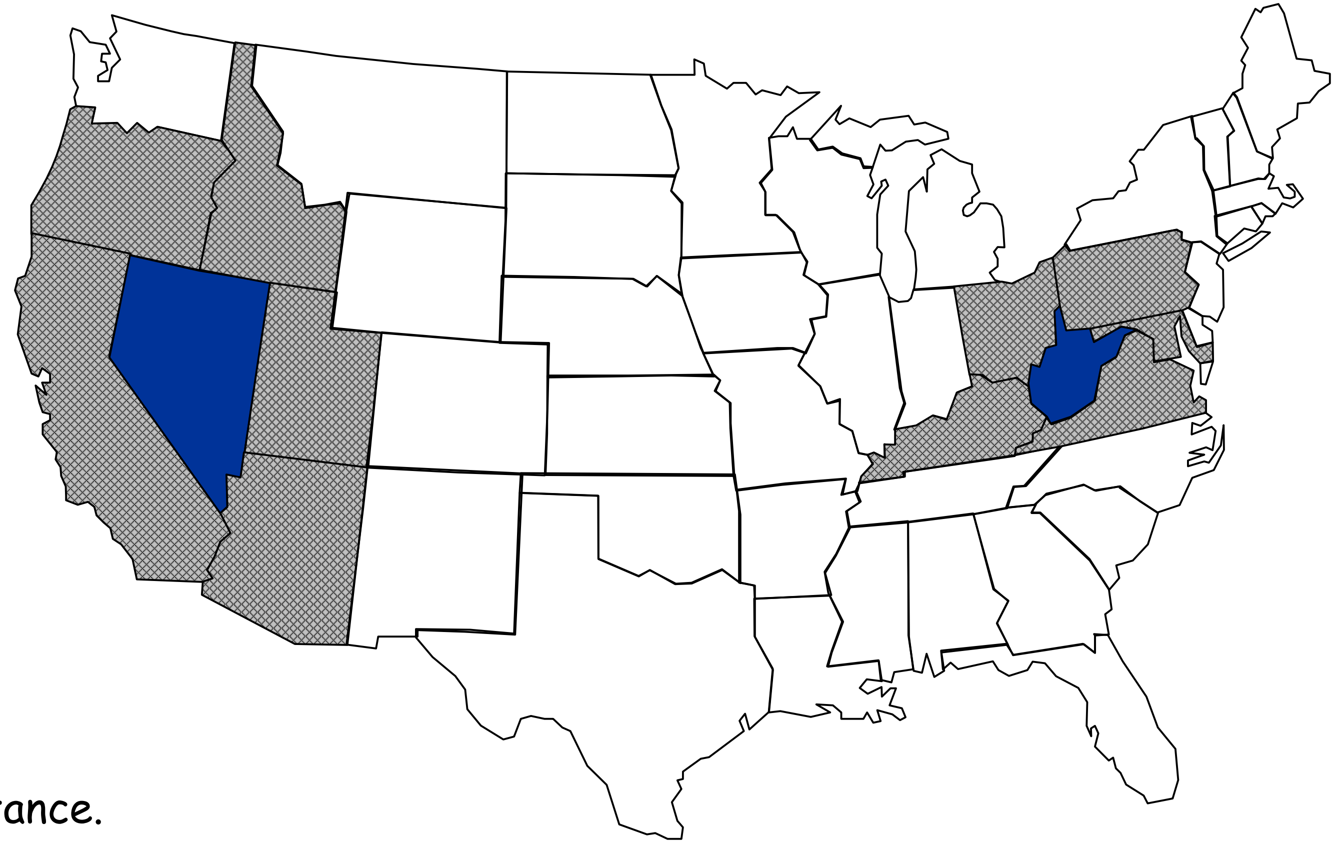
PLANAR-3-COLOR. Given a planar map, can it be colored using 3 colors so that no adjacent regions have the same color?



YES instance.

Planar 3-Colorability

PLANAR-3-COLOR. Given a planar map, can it be colored using 3 colors so that no adjacent regions have the same color?

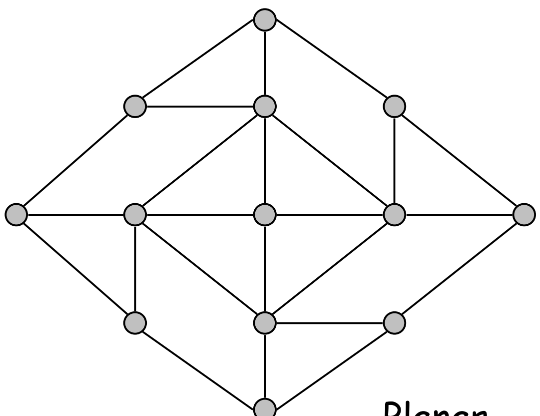


NO instance.

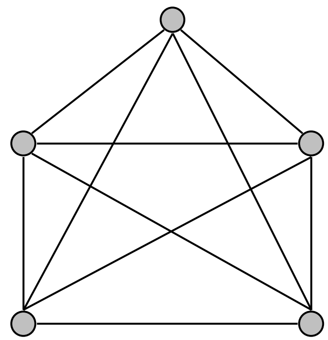
Planarity

Def. A graph is planar if it can be embedded in the plane in such a way that no two edges cross.

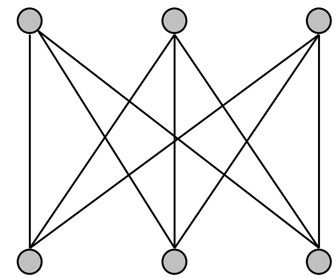
Applications: VLSI circuit design, computer graphics.



Planar



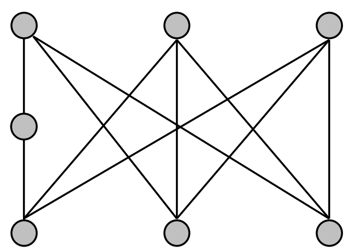
K_5: non-planar



$K_{3,3}$: non-planar

Kuratowski's Theorem. An undirected graph G is non-planar iff it contains a subgraph homeomorphic to K_5 or $K_{3,3}$.

homeomorphic to $K_{3,3}$ →



Planarity Testing

Planarity testing. [Hopcroft-Tarjan 1974] $O(n)$.

simple planar graph can have at most 3n edges

Remark. Many intractable graph problems can be solved in poly-time if the graph is planar; many tractable graph problems can be solved faster if the graph is planar.

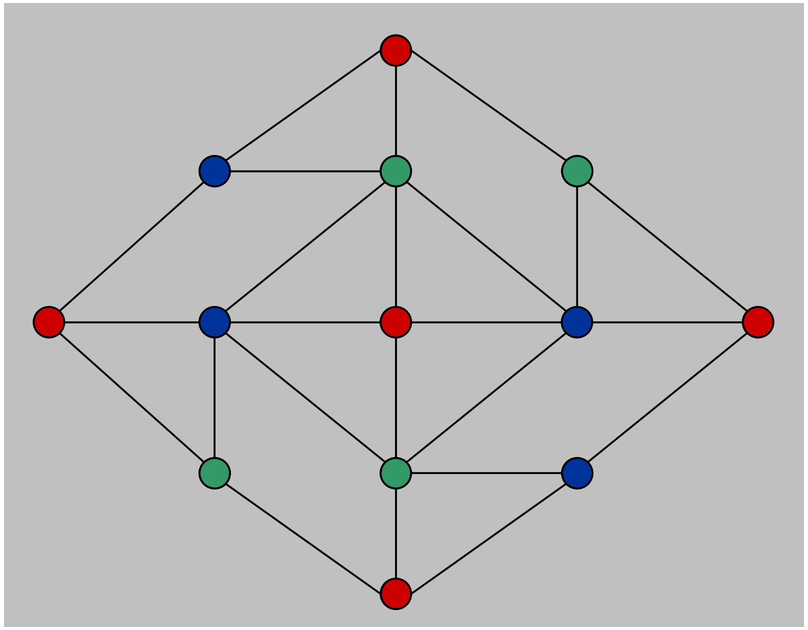
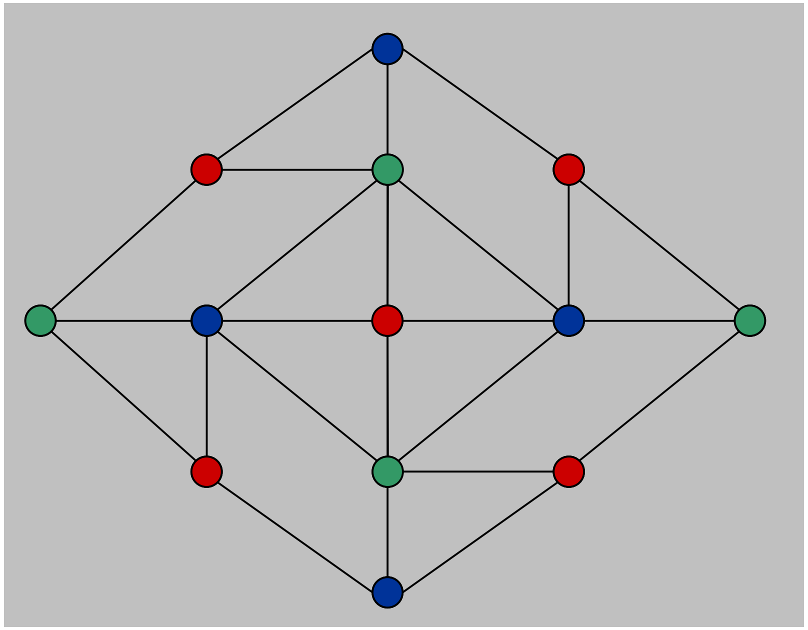
Planar 3-Colorability

Claim. 3-COLOR $\leq_P$ PLANAR-3-COLOR.

Proof sketch: Given instance of 3-COLOR, draw graph in plane, letting edges cross if necessary.

Replace each edge crossing with the following planar gadget W.

- in any 3-coloring of W, opposite corners have the same color
- any assignment of colors to the corners in which opposite corners have the same color extends to a 3-coloring of W

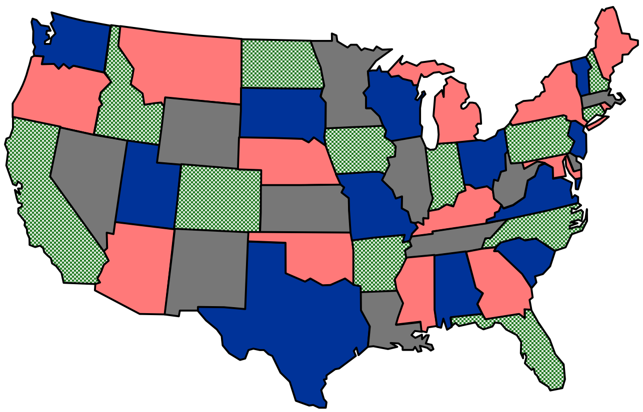



Planar k-Colorability

PLANAR-2-COLOR. Solvable in linear time.

PLANAR-3-COLOR. NP-complete.

PLANAR-4-COLOR. Solvable in O(1) time.



Theorem. [Appel-Haken, 1976] Every planar map is 4-colorable.

- Resolved century-old open problem.
- Used 50 days of computer time to deal with many special cases.
- First major theorem to be proved using computer.

False intuition. If PLANAR-3-COLOR is hard, then so is PLANAR-4-COLOR and PLANAR-5-COLOR.

Polynomial-Time Detour

Graph minor theorem. [Robertson-Seymour 1980s]

Corollary. There exist an $O(n^3)$ algorithm to determine if a graph can be embedded in the torus圆环面 in such a way that no two edges cross.

Pf of theorem. Tour de force绝技.

Polynomial-Time Detour

Graph minor theorem. [Robertson-Seymour 1980s]

Corollary. There exist an $O(n^3)$ algorithm to determine if a graph can be embedded in the torus in such a way that no two edges cross.

Mind boggling fact 1. The proof is highly non-constructive!
Mind boggling fact 2. The constant of proportionality is enormous!

> Unfortunately, for any instance $G = (V, E)$ that one could fit into the known universe, one would easily prefer n^{70} to even *constant* time, if that constant had to be one of Robertson and Seymour's. - *David Johnson*

Theorem. There exists an explicit $O(n)$ algorithm.
Practice. LEDA implementation guarantees $O(n^3)$.

Homeworks

Homework: 8.3, 8.7, 8.16,8.39

3. Suppose you're helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who's skilled at each of the n sports covered by the camp (baseball, volleyball, and so on). They have received job applications from m potential counselors. For each of the n sports, there is some subset of the m applicants qualified in that sport. The question is: For a given number $k < m$, is it possible to hire at most k of the counselors and have at least one counselor qualified in each of the n sports? We'll call this the *Efficient Recruiting Problem*.

Show that Efficient Recruiting is NP-complete.

7. Since the 3-Dimensional Matching Problem is NP-complete, it is natural to expect that the corresponding 4-Dimensional Matching Problem is at least as hard. Let us define *4-Dimensional Matching* as follows. Given sets W, X, Y, and Z, each of size n, and a collection C of ordered 4-tuples of the form (w_i, x_j, y_k, z_ℓ), do there exist n 4-tuples from C so that no two have an element in common?

Prove that 4-Dimensional Matching is NP-complete.

16. Consider the problem of reasoning about the identity of a set from the size of its intersections with other sets. You are given a finite set U of size n, and a collection $A_1, \ldots, A_m$ of subsets of U. You are also given numbers $c_1, \ldots, c_m$. The question is: Does there exist a set $X \subset U$ so that for each $i = 1, 2, \ldots, m$, the cardinality of $X \cap A_i$ is equal to c_i? We will call this an instance of the *Intersection Inference Problem*, with input U, $\{A_i\}$, and $\{c_i\}$.

Prove that Intersection Inference is NP-complete.

39. The *Directed Disjoint Paths Problem* is defined as follows. We are given a directed graph G and k pairs of nodes $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. The problem is to decide whether there exist node-disjoint paths $P_1, P_2, \ldots, P_k$ so that P_i goes from s_i to t_i.

Show that Directed Disjoint Paths is NP-complete.