



CMU 15-251

REDUCTIONS

TEACHERS:

ANIL ADA

ARIEL PROCACCIA (THIS TIME)

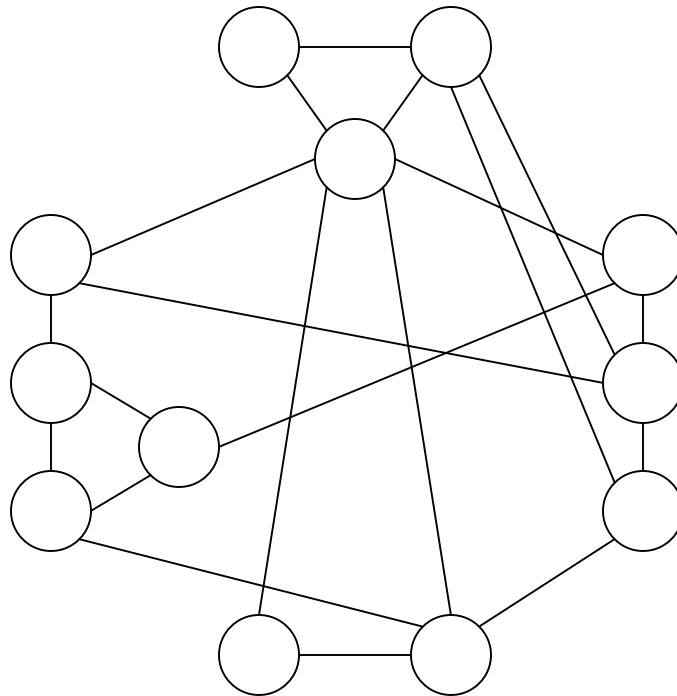
k -COLORING

- Reminder: a k -coloring of a graph satisfies:
 - Each node has a color
 - There are at most k different colors
 - Every two nodes connected by an edge have different colors
- A graph is k -colorable iff it has a k -coloring



2-COLORING

- Is this graph 2-colorable?



2-COLORING

- Given a graph G , how can we decide if it is 2-colorable?
- Enumerate all possible 2^n colorings to look for a valid one...
- OK, but how can we **efficiently** decide if G is 2-colorable?
 - In polynomial time in the number of vertices n



2-COLORING

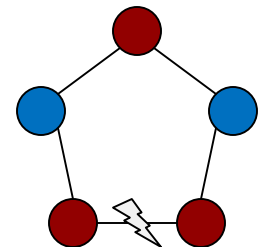
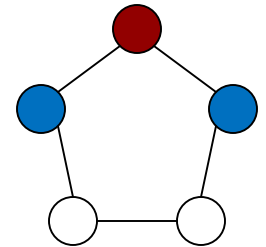
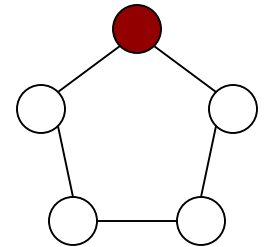
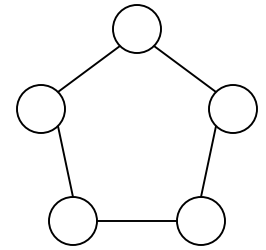
- **Poll 1:** $G = (V, E)$ is 2-colorable iff:
 1. G has a Hamiltonian cycle
 2. $|E| \leq |V| - 1$
 3. Every vertex in G has even degree
 - ④ G has no odd cycles

What's another
term for
2-colorable?



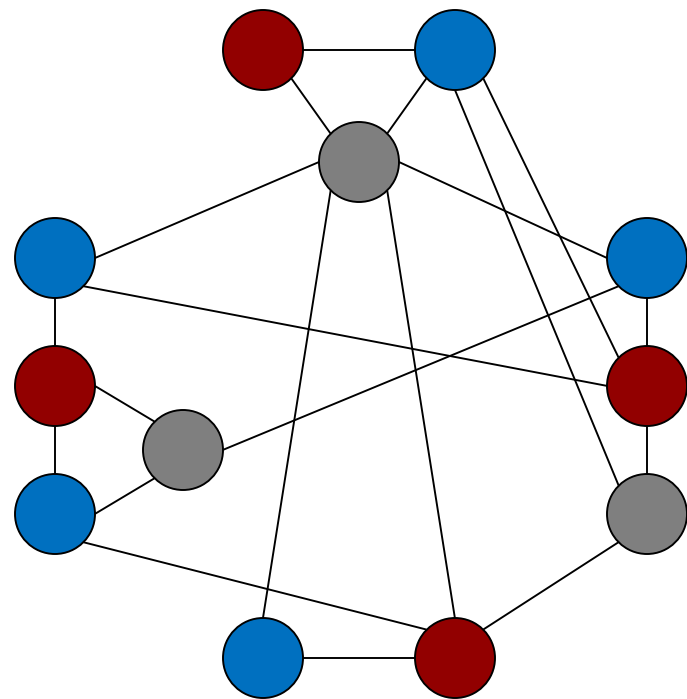
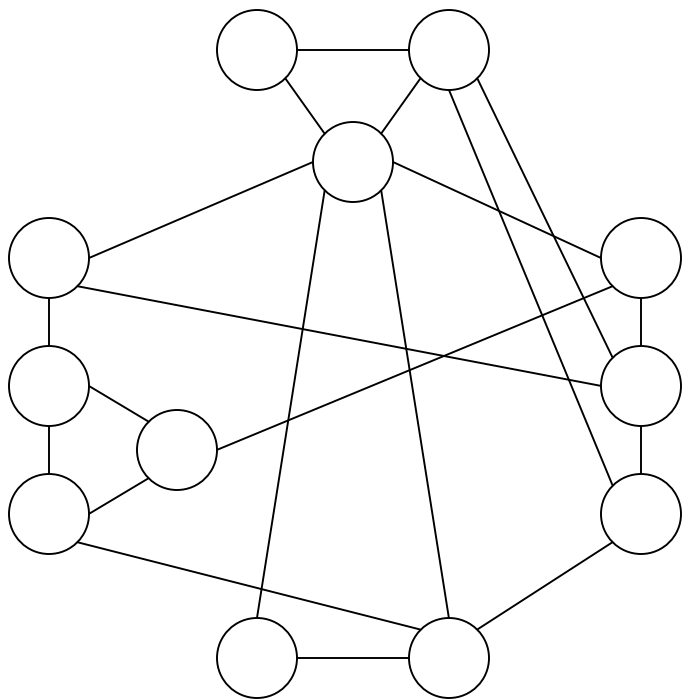
2-COLORING

- Algorithm:
 - Choose an arbitrary node, color it red and its neighbors blue
 - Color the uncolored neighbors of the blue vertices red, etc.
 - If G is not connected, repeat for every component



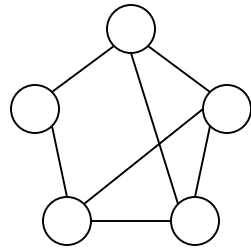
3-COLORING

- Is this graph 3-colorable?

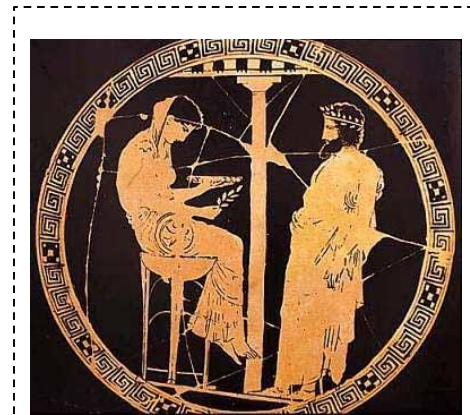


3-COLORABILITY ORACLES

- We can decide 3-colorability by trying all 3^n possible colorings
- Let's say we can ask an oracle...



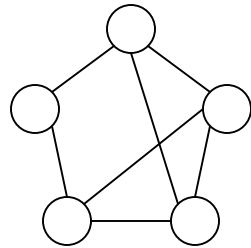
NO / YES



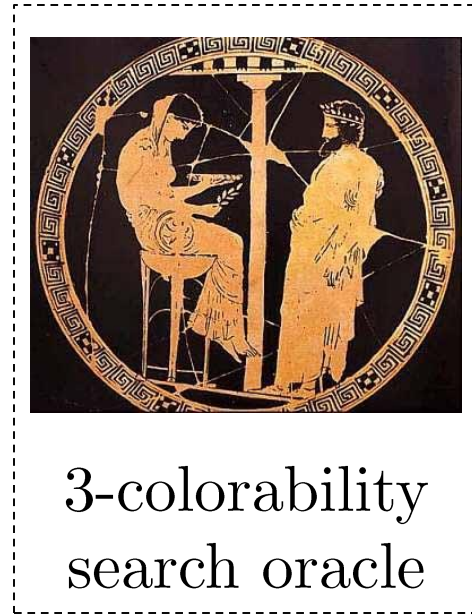
3-colorability
decision oracle

3-COLORABILITY ORACLES

- How do we turn a decision oracle into a search oracle?



NO / YES, here's how

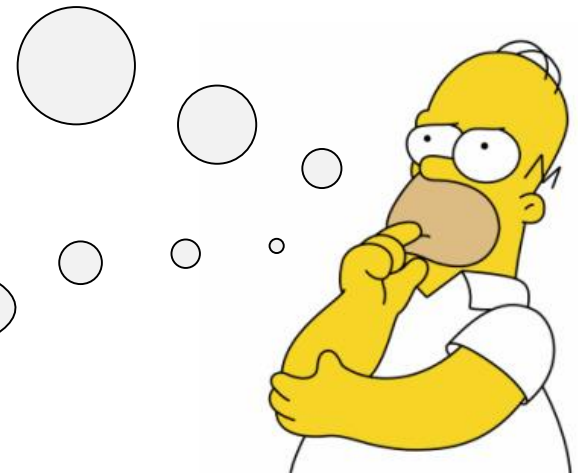


3-colorability
search oracle

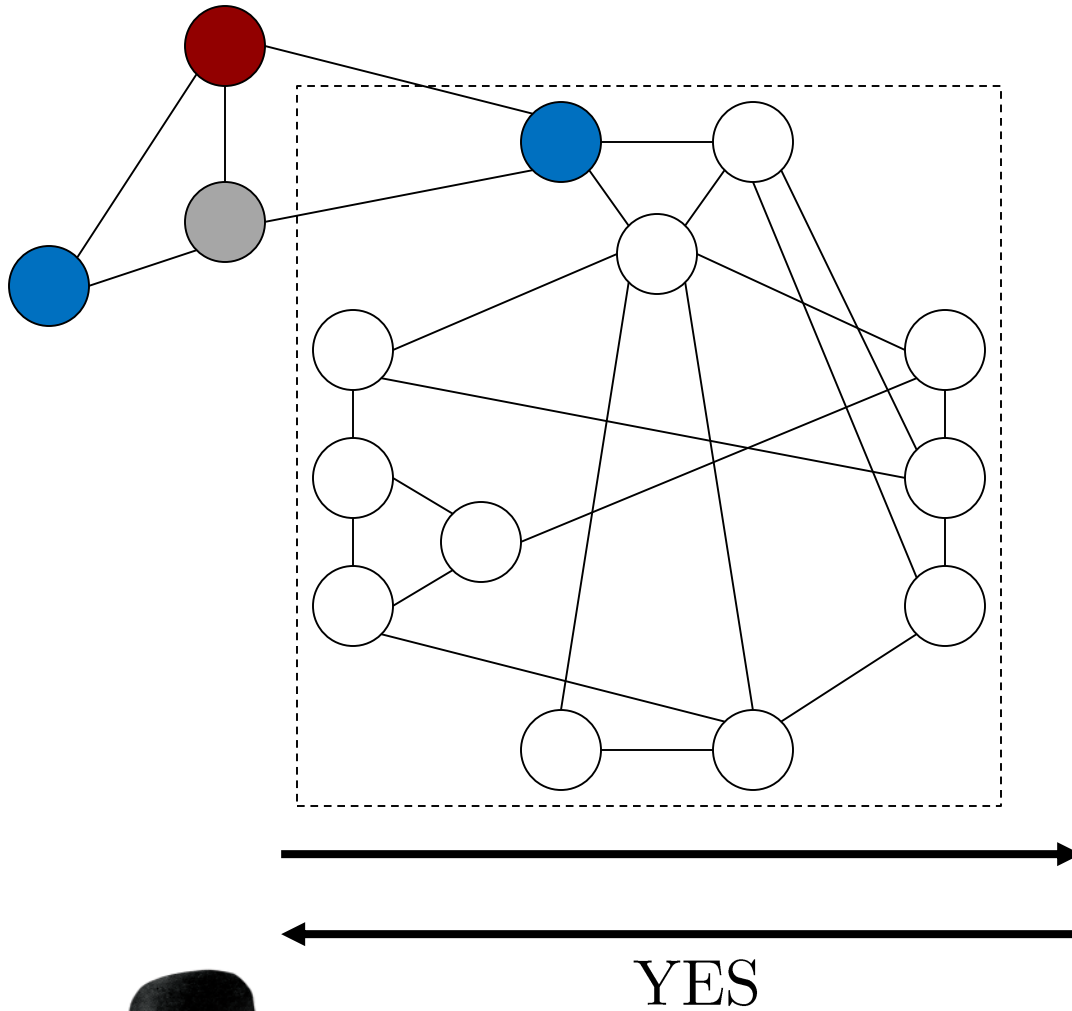
3-COLORABILITY ORACLES

What if I gave the oracle partial colorings of G ? For each partial coloring of G , I could pick an uncolored node and try different colors on it until the oracle says "YES". I would then have a larger partial coloring

The oracle doesn't accept partial colorings!

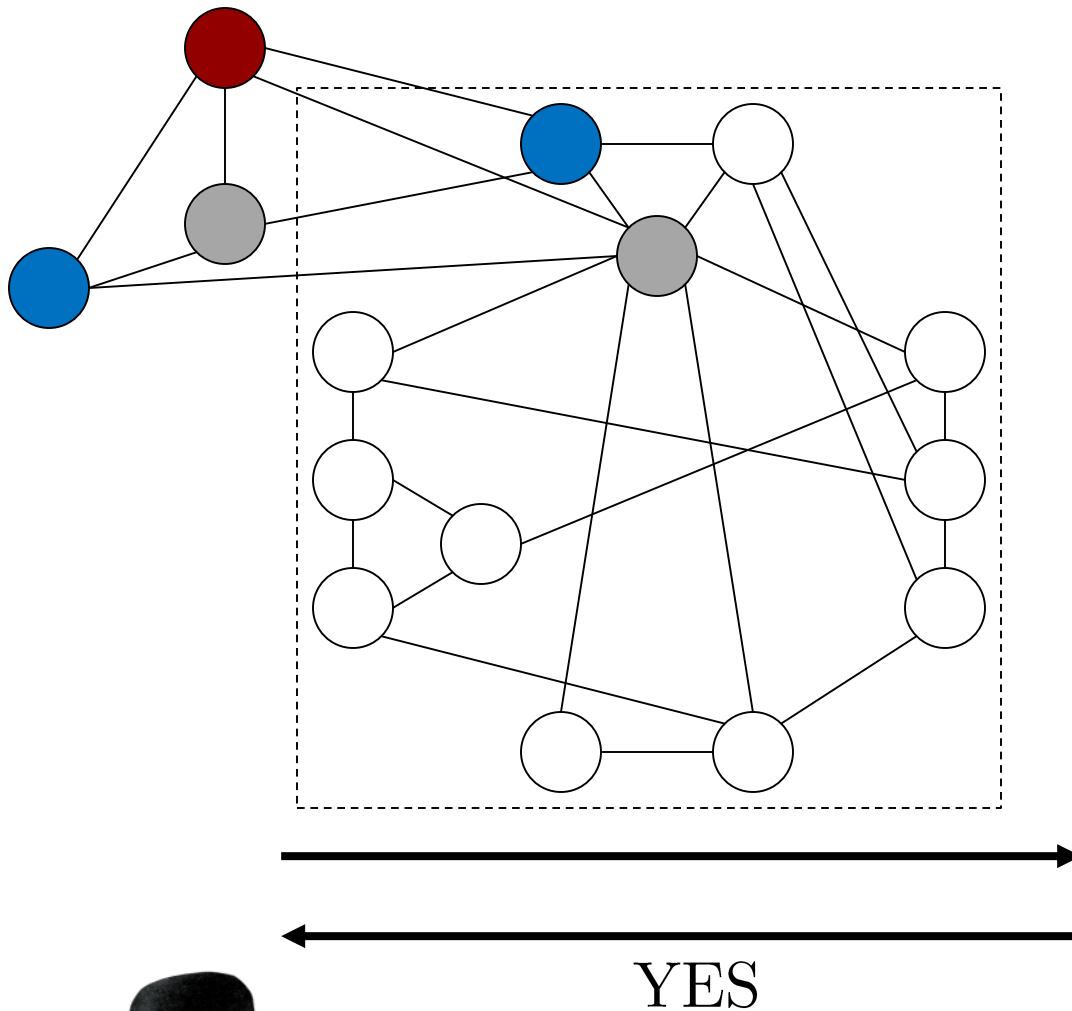


3-COLORABILITY ORACLES



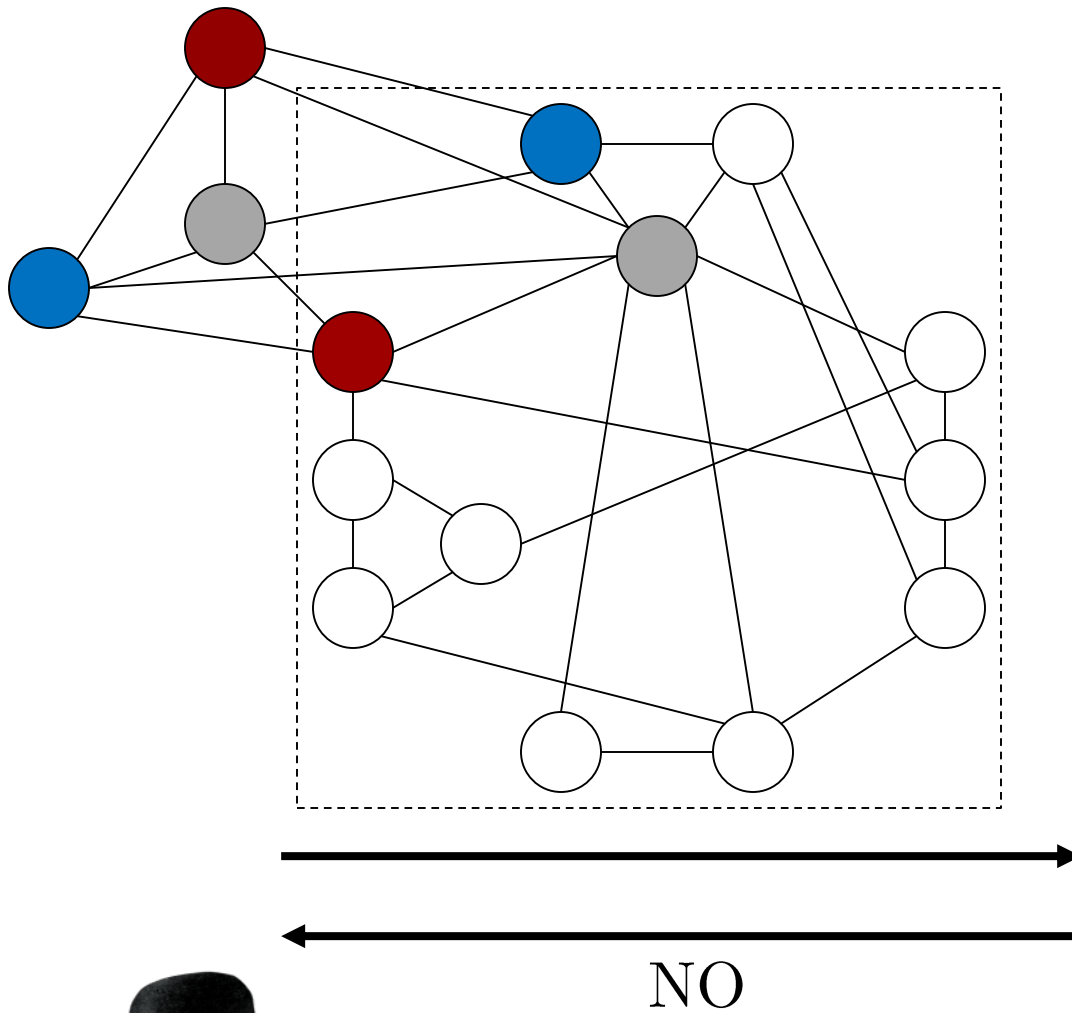
Given:
3-colorability
decision oracle

3-COLORABILITY ORACLES



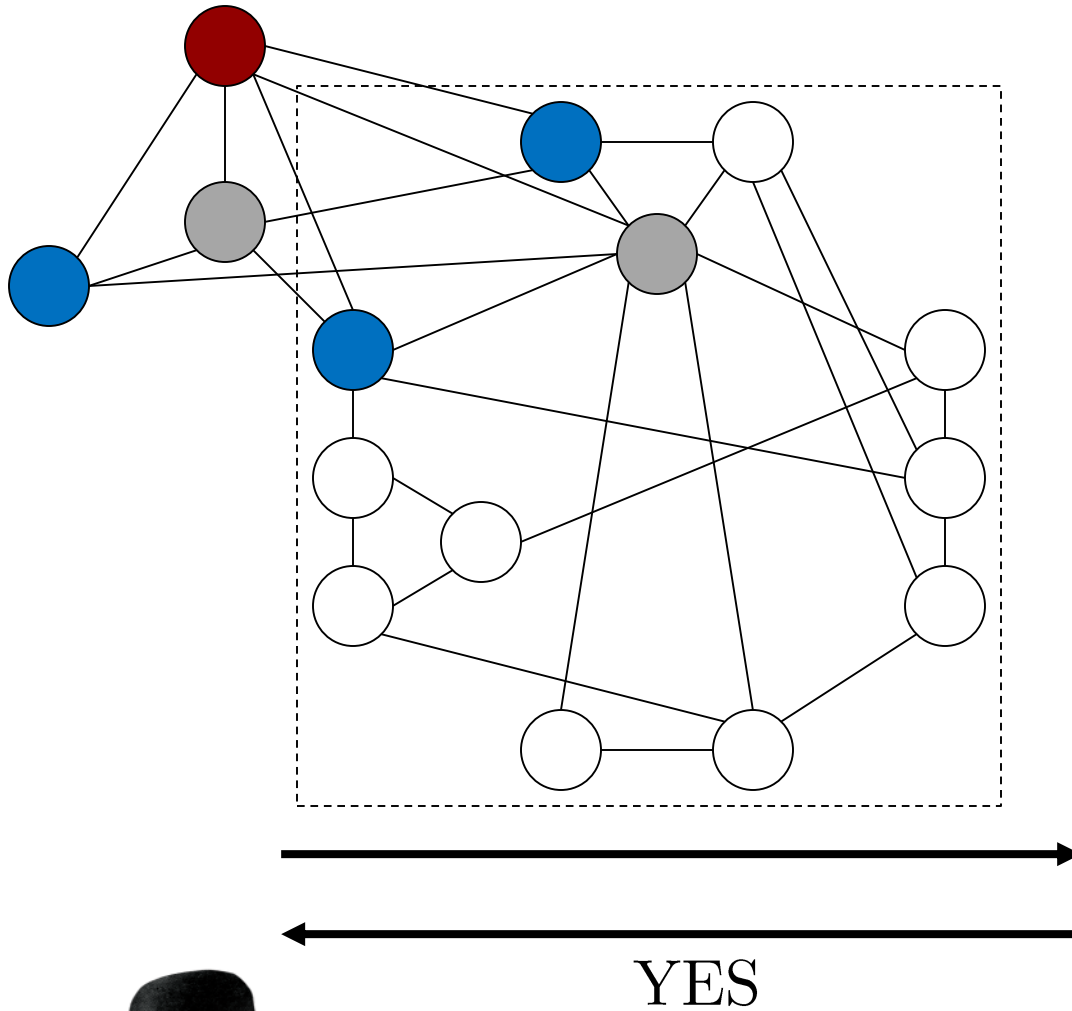
Given:
3-colorability
decision oracle

3-COLORABILITY ORACLES



Given:
3-colorability
decision oracle

3-COLORABILITY ORACLES



Given:
3-colorability
decision oracle

3-COLORABILITY ORACLES

A 3-colorability search oracle can be simulated using a **linear** number of calls to a decision oracle!

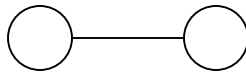


CLIQUE

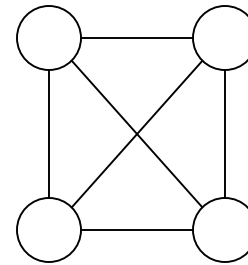
- Reminder: A *k*-clique is a set of *k* nodes with all possible edges between them



1-clique



2-clique

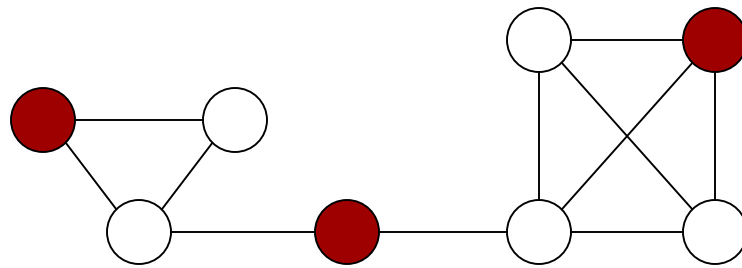


4-clique

- **CLIQUE:** Given a graph G and $k \in \mathbb{N}$, does G contain a *k*-clique?

INDEPENDENT SET

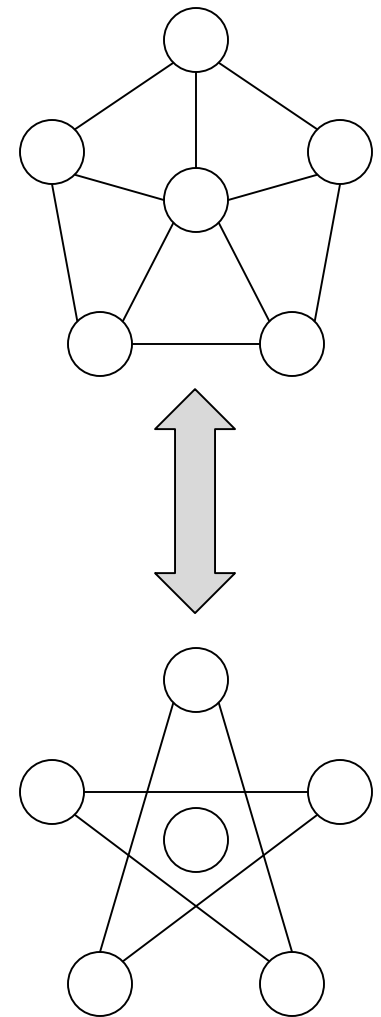
- A *k*-independent set is a set of *k* nodes with no edges between them



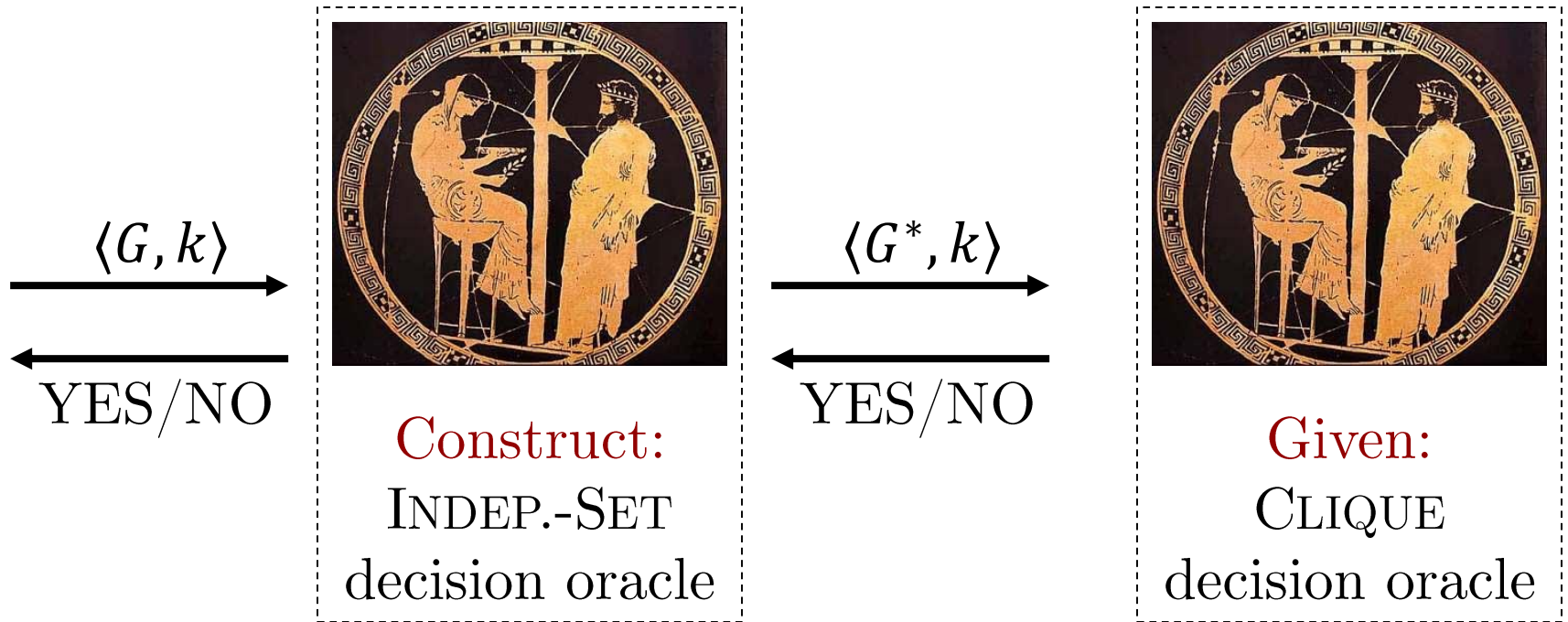
- **INDEPENDENT-SET:** Given a graph G and $k \in \mathbb{N}$, does G contain a k -independent set?

CLIQUE VS. IS

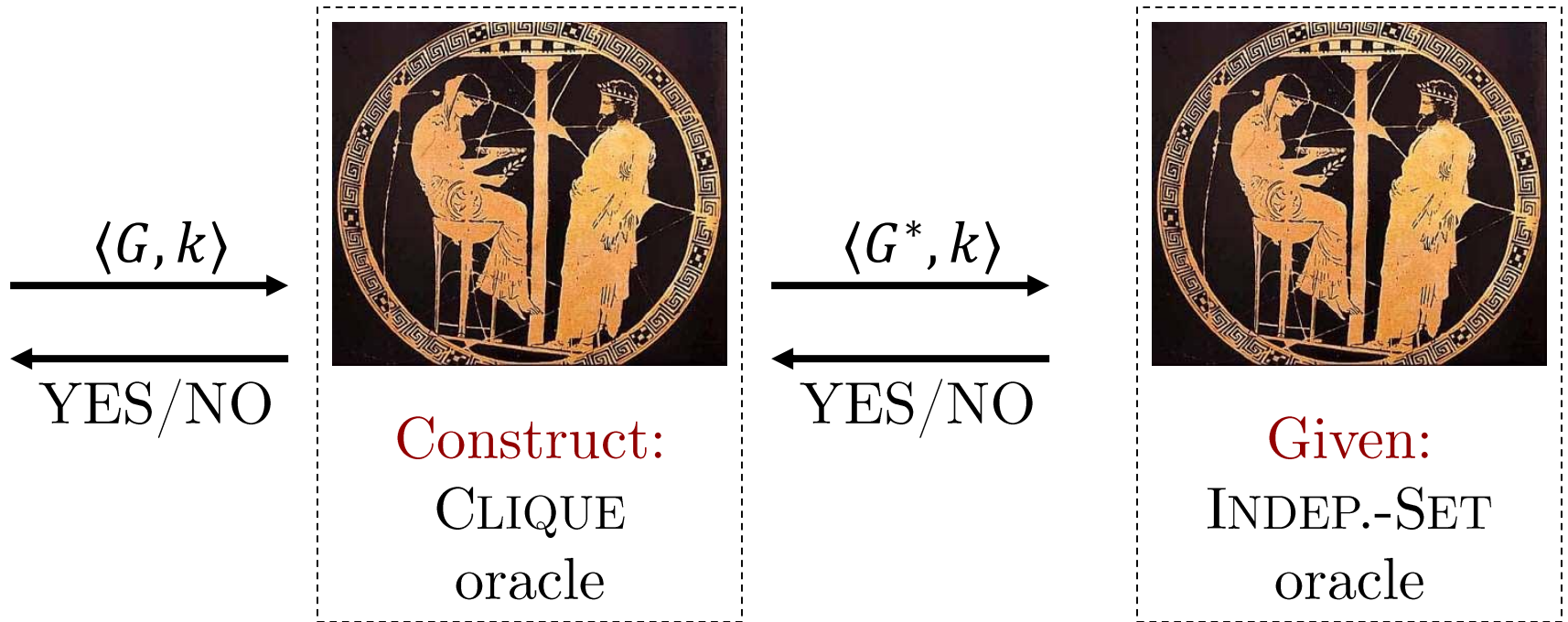
- Let $G^* = (V, E^*)$ be the complement of $G = (V, E)$
 $(u, v) \in E \Leftrightarrow (u, v) \notin E^*$
- **Poll 2:** G has a k -clique for $k \geq 2$ iff:
 1. G^* has an IS of size $k - 2$
 2. G^* has an IS of size $k - 1$
 3. G^* has an IS of size k
 4. G^* has an IS of size $k + 1$



CLIQUE VS. IS



CLIQUE VS. IS



CLIQUE VS. IS

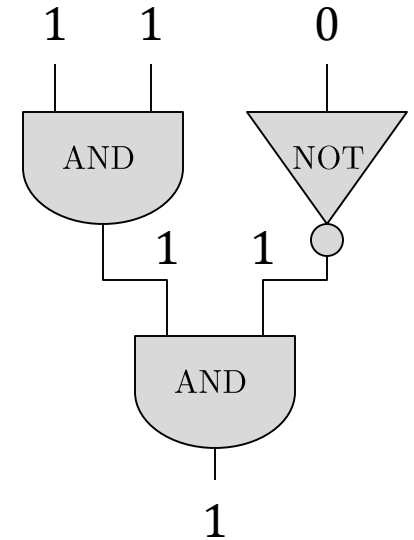
- We can quickly reduce an instance of CLIQUE to an instance of INDEPENDENT-SET, and vice versa
- There is a fast method for one iff there is a fast method for the other

CLIQUE and INDEPENDENT-SET are cosmetically different but essentially the same!

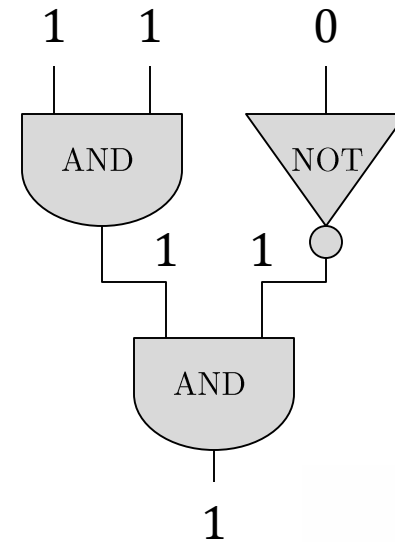
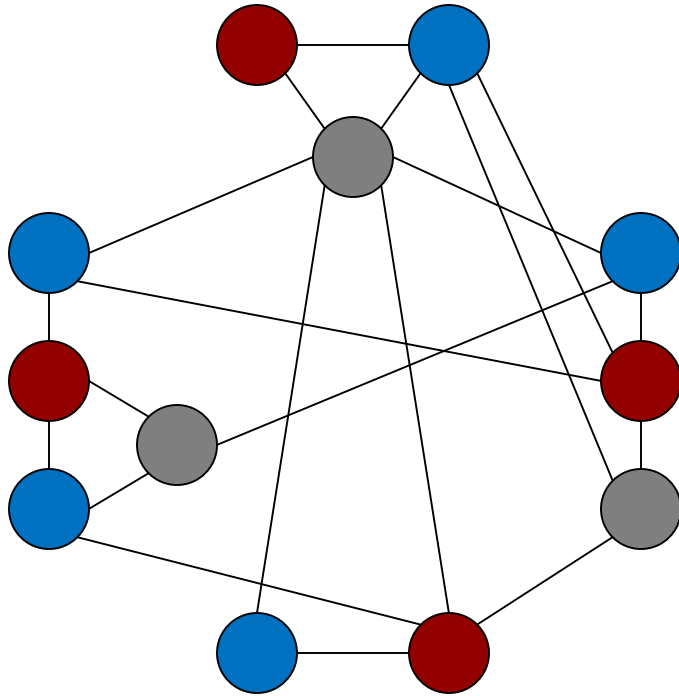


CIRCUIT-SAT

- AND, OR, NOT gates wired together
- **CIRCUIT-SATISFIABILITY:** Given a circuit with n inputs and one output, is there a way to assign 0/1 values to the input wires so that the output value is 1 (true)?



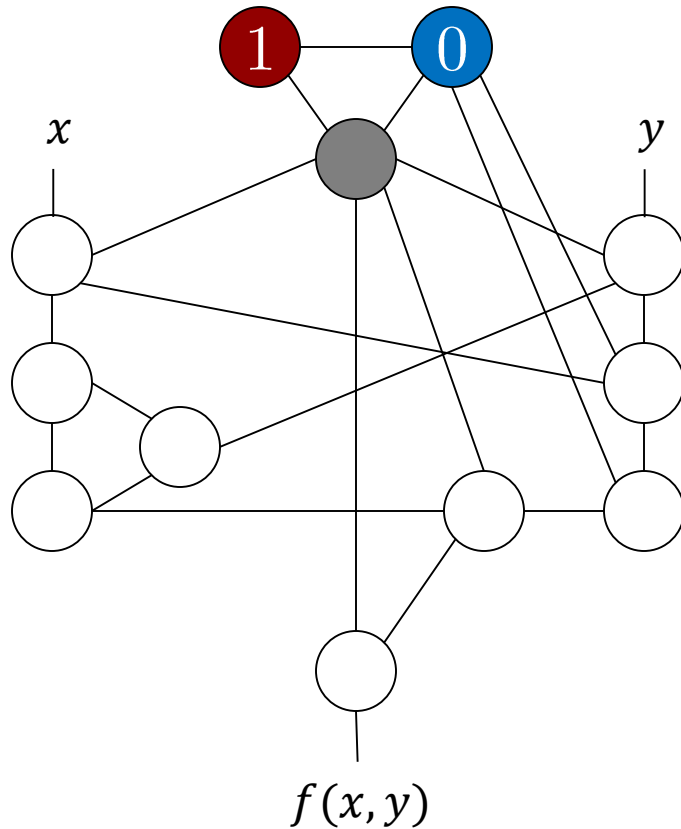
3-COLORABILITY VS. CIRCUIT-SAT



These two problems are fundamentally different!

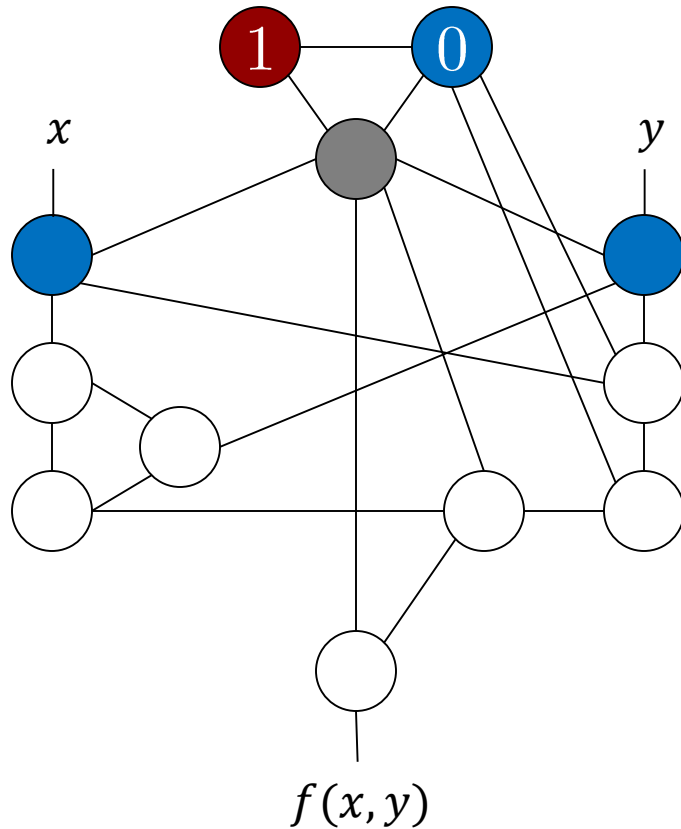


3-COLORABILITY VS. CIRCUIT-SAT



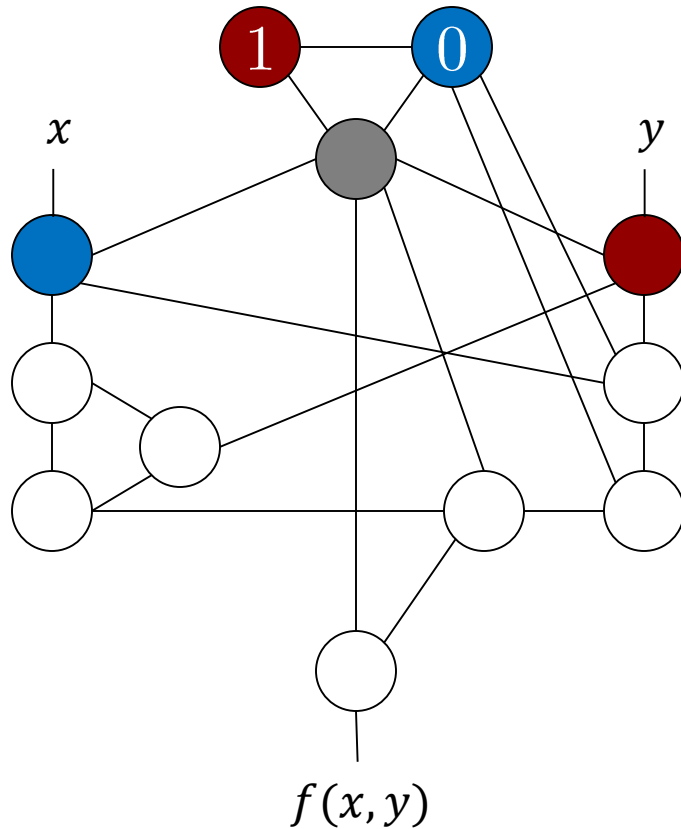
x	y	$f(x, y)$

3-COLORABILITY VS. CIRCUIT-SAT



x	y	$f(x, y)$
0	0	

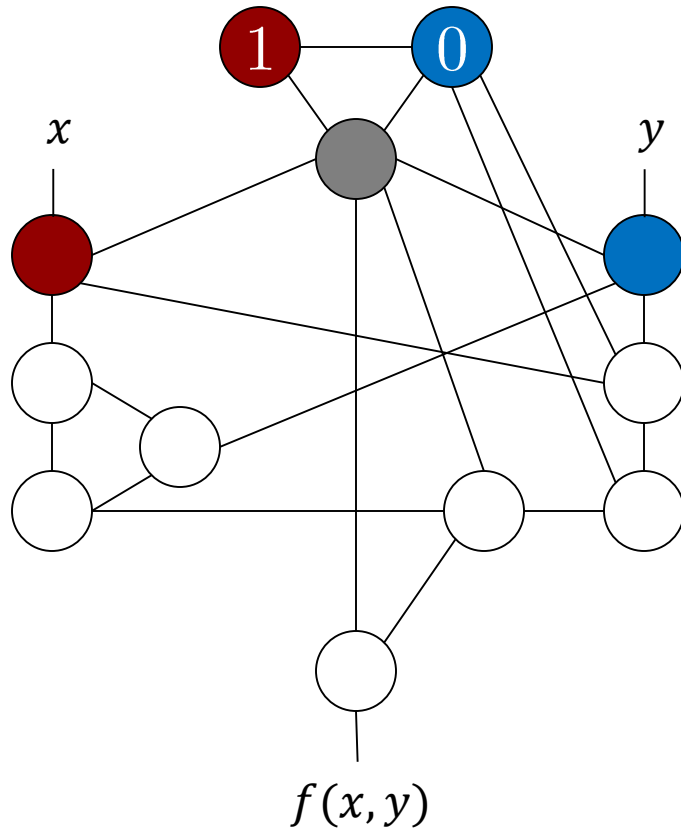
3-COLORABILITY VS. CIRCUIT-SAT



x	y	$f(x, y)$
0	0	0
0	1	

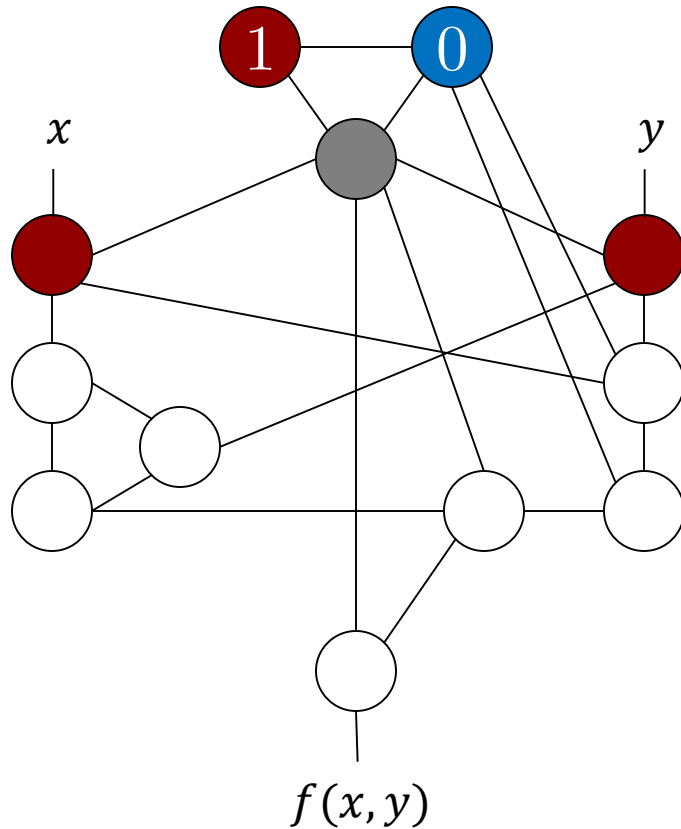


3-COLORABILITY VS. CIRCUIT-SAT



x	y	$f(x, y)$
0	0	0
0	1	1
1	0	

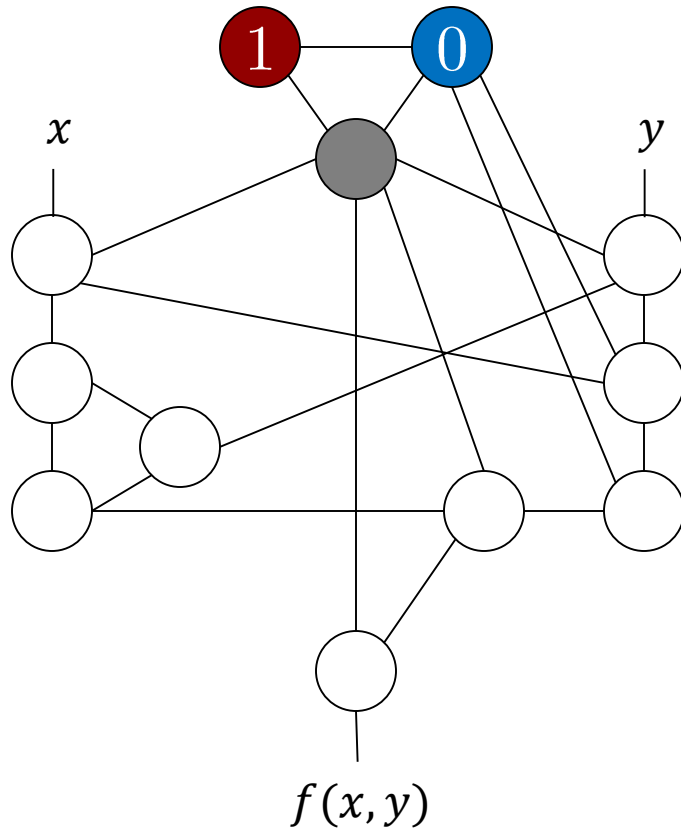
3-COLORABILITY VS. CIRCUIT-SAT



x	y	$f(x, y)$
0	0	0
0	1	1
1	0	1
1	1	

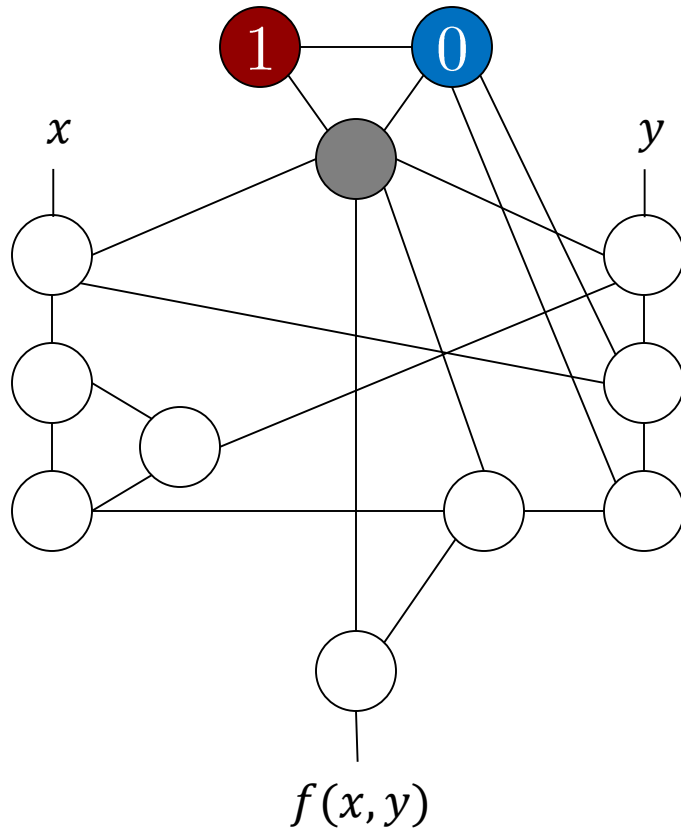


3-COLORABILITY VS. CIRCUIT-SAT



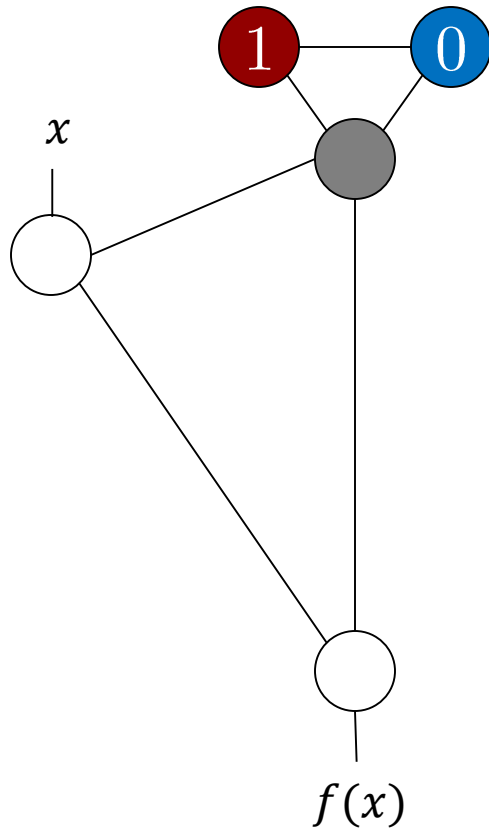
x	y	$f(x, y)$
0	0	0
0	1	1
1	0	1
1	1	1

3-COLORABILITY VS. CIRCUIT-SAT



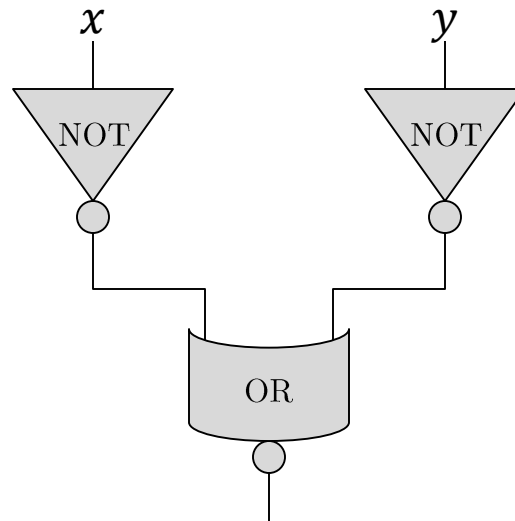
x	y	OR
0	0	0
0	1	1
1	0	1
1	1	1

3-COLORABILITY VS. CIRCUIT-SAT



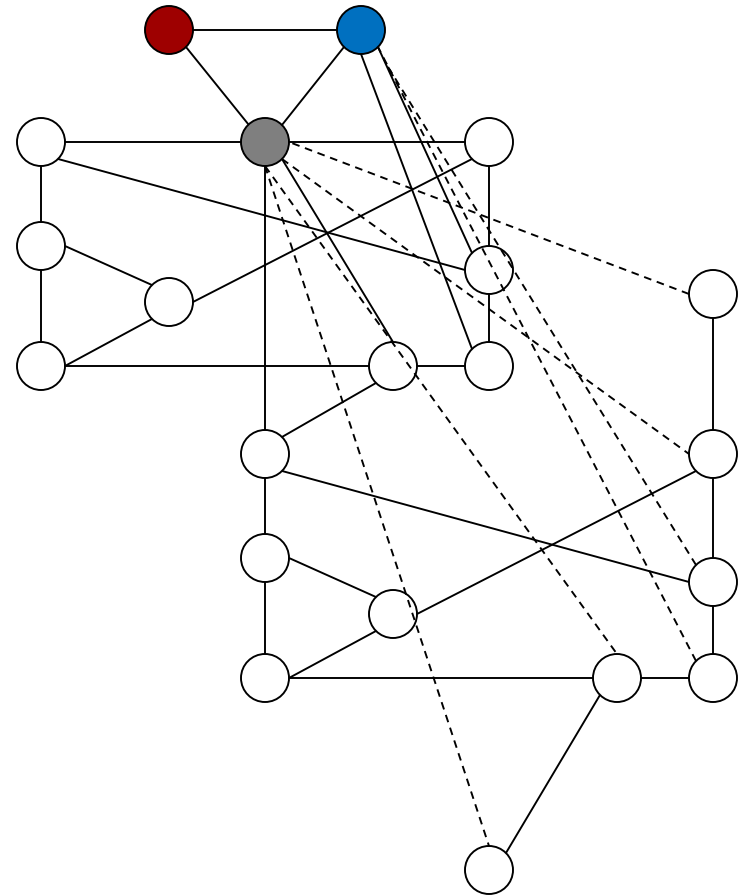
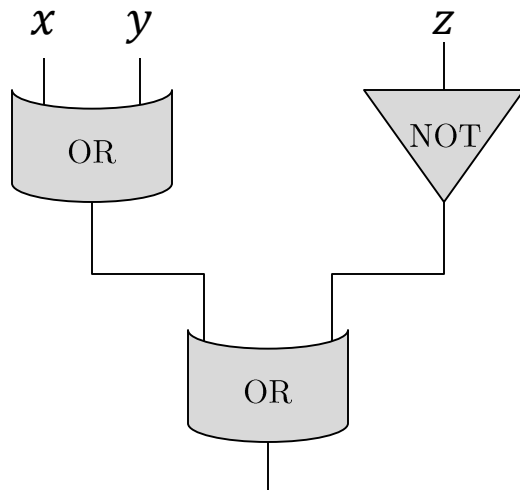
x	NOT
0	1
1	0

3-COLORABILITY VS. CIRCUIT-SAT

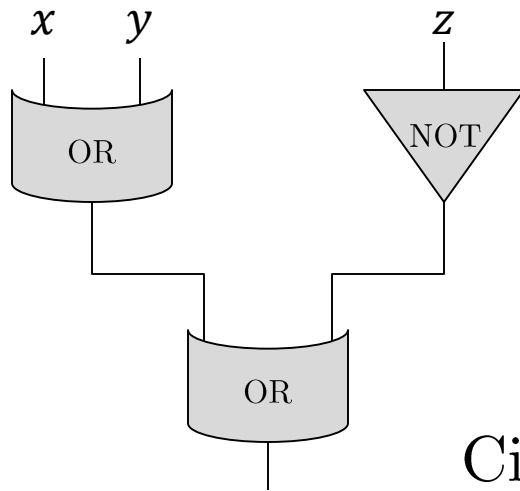


AND Gate from OR and NOT

3-COLORABILITY VS. CIRCUIT-SAT



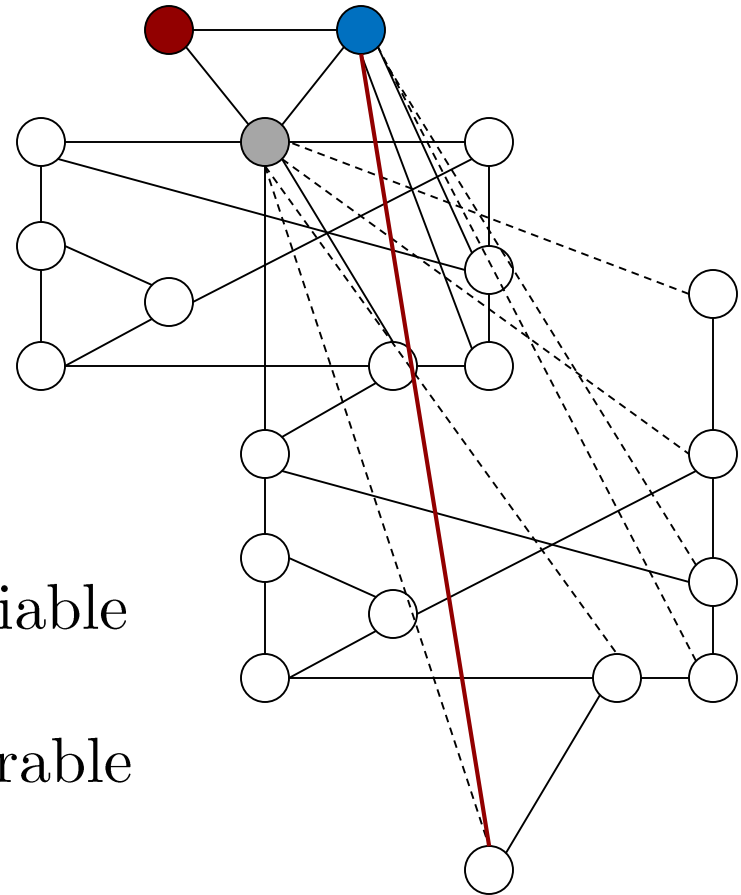
3-COLORABILITY VS. CIRCUIT-SAT



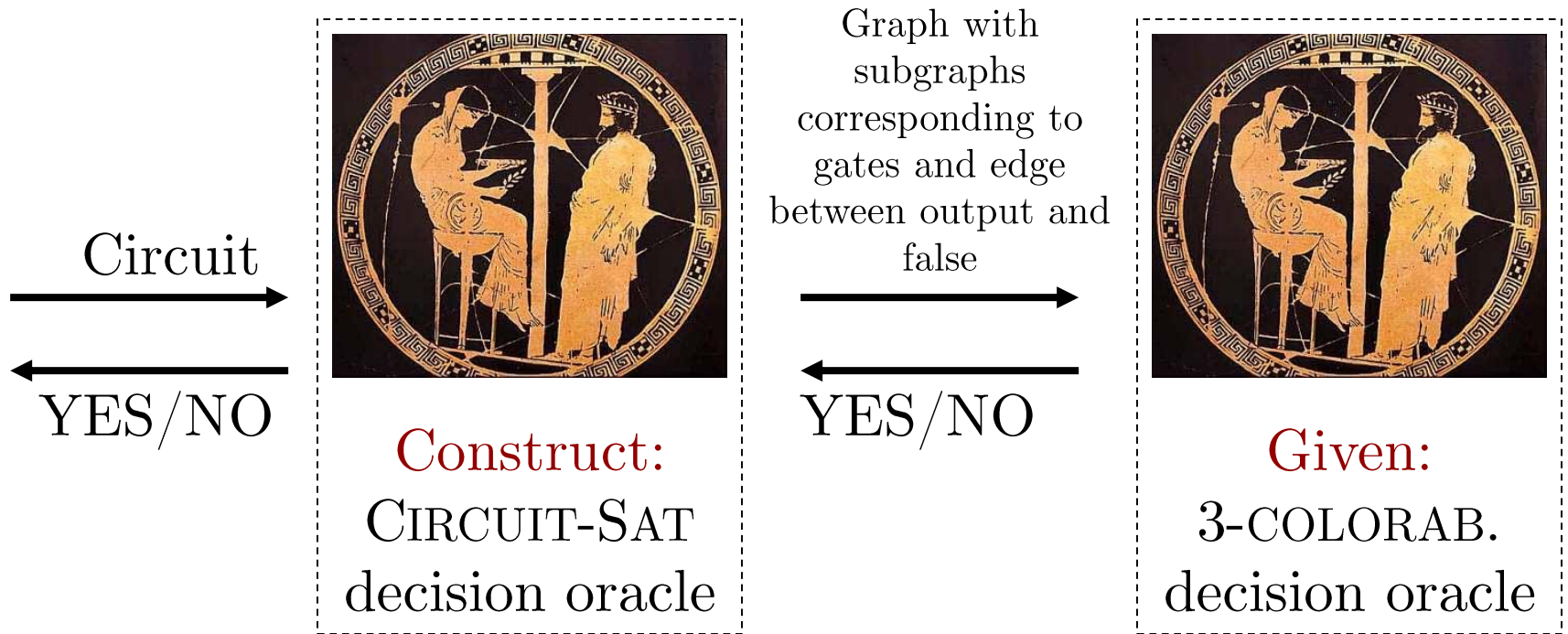
Circuit is satisfiable



Graph is 3-colorable



3-COLORABILITY VS. CIRCUIT-SAT



3-COLORABILITY VS. CIRCUIT-SAT

- There is a linear-time function that reduces instances of CIRCUIT-SAT to instances of 3-COLORABILITY
- **Fact:** There are efficient ways to reduce an instance of any of the four problems we discussed to an instance of any other

But nobody knows how to efficiently solve any of these four problems in the worst case!



WHAT WE HAVE LEARNED

- Definitions:
 - k -COLORING
 - CLIQUE
 - INDEPENDENT-SET
 - CIRCUIT-SAT
- Principles:
 - Computationally efficient reductions between problems!

