6.S890:
Topics in Multiagent Learning

Lecture 7 – Prof. Daskalakis

Fall 2023

Reminder:
Nash Equilibrium Existence and Computation

[John Nash '50]: A Nash equilibrium exists in **every** finite game.

Deep influence in Economics, enabling other existence results.

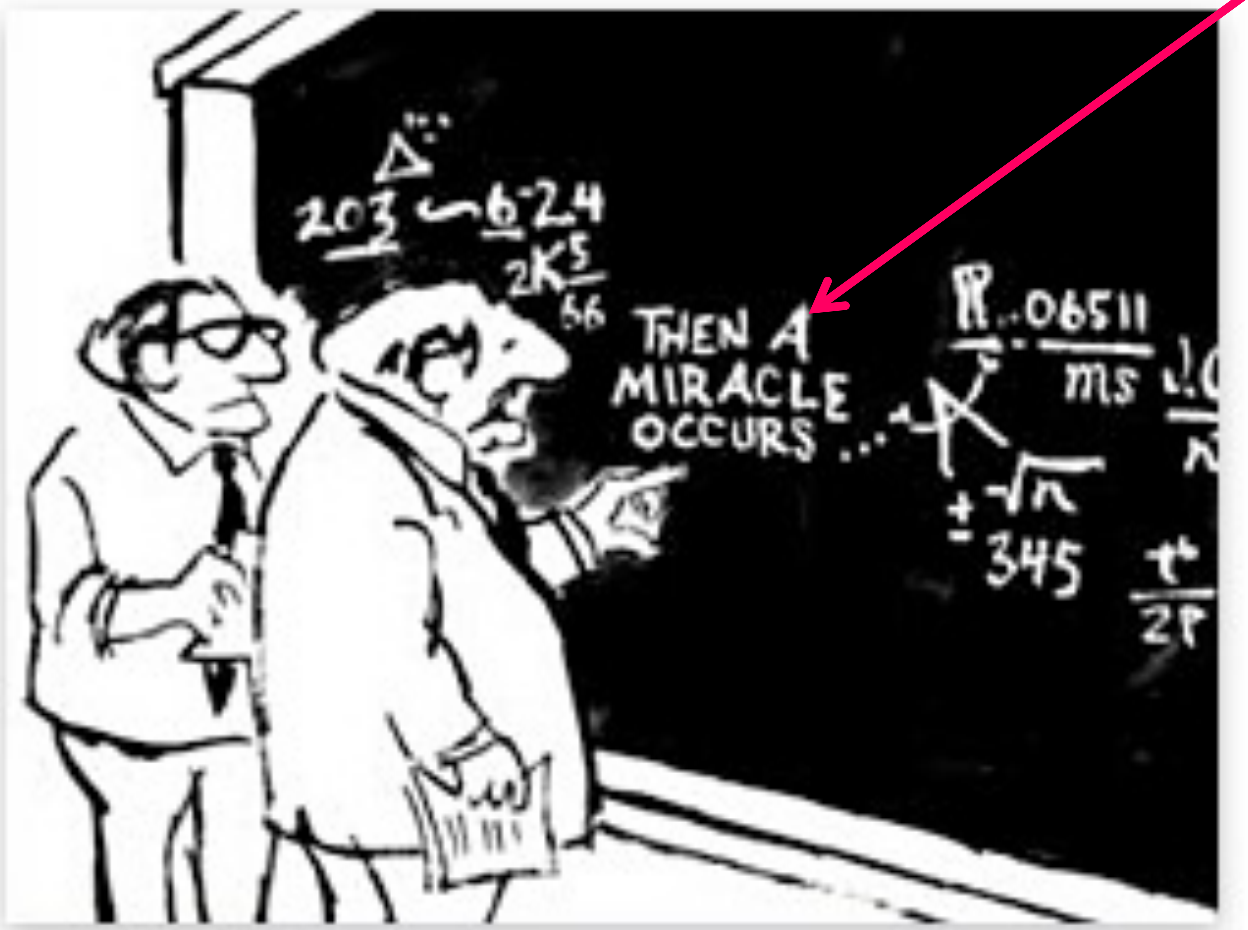
Proof non-constructive (uses Brouwer's fixed point theorem)

No simpler proof has been discovered, and no polynomial-time algorithm has been discovered either.

Towards understanding why this is, we looked deeper in the proof of Nash's theorem

What step in Nash's existence proof cannot be executed in polynomial-time?

??

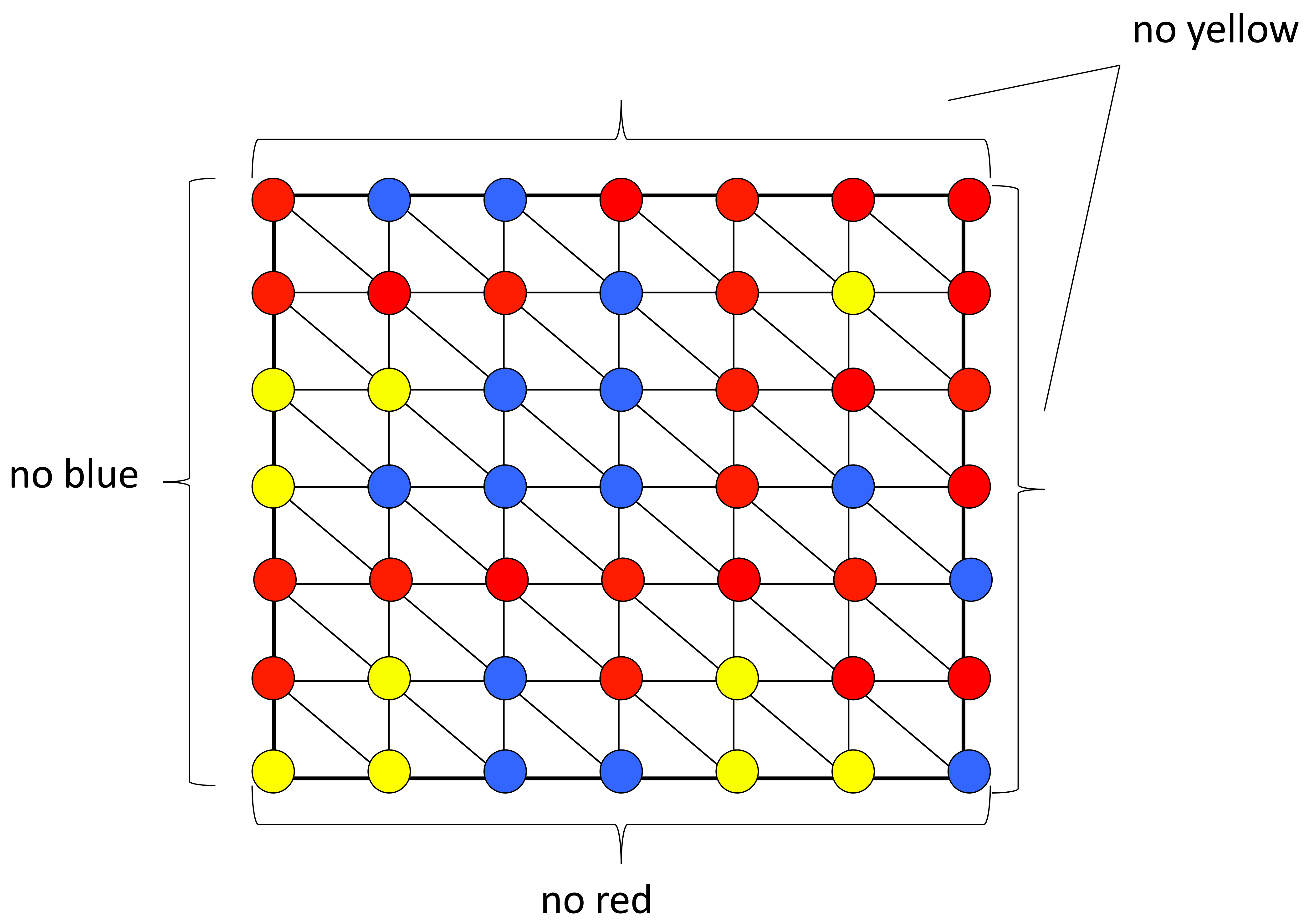


A Chain of Implications

Nash Theorem $\Leftarrow$ Brouwer Theorem $\Leftarrow$ Sperner Lemma

- [Nash 1950]: Every finite game has a Nash equilibrium.
- [Brouwer 1911]: Every continuous function $f: D \to D$ from a convex compact set D to itself has a fixed point $x^* = f(x^*)$.
- [Sperner 1928]: Every legal 3-coloring of a 2-d triangulated square has a tri-chromatic triangle.

Sperner's Lemma (in 2-d)



[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.

A Chain of Implications

Nash Theorem $\Leftarrow$ Brouwer Theorem $\Leftarrow$ Sperner Lemma

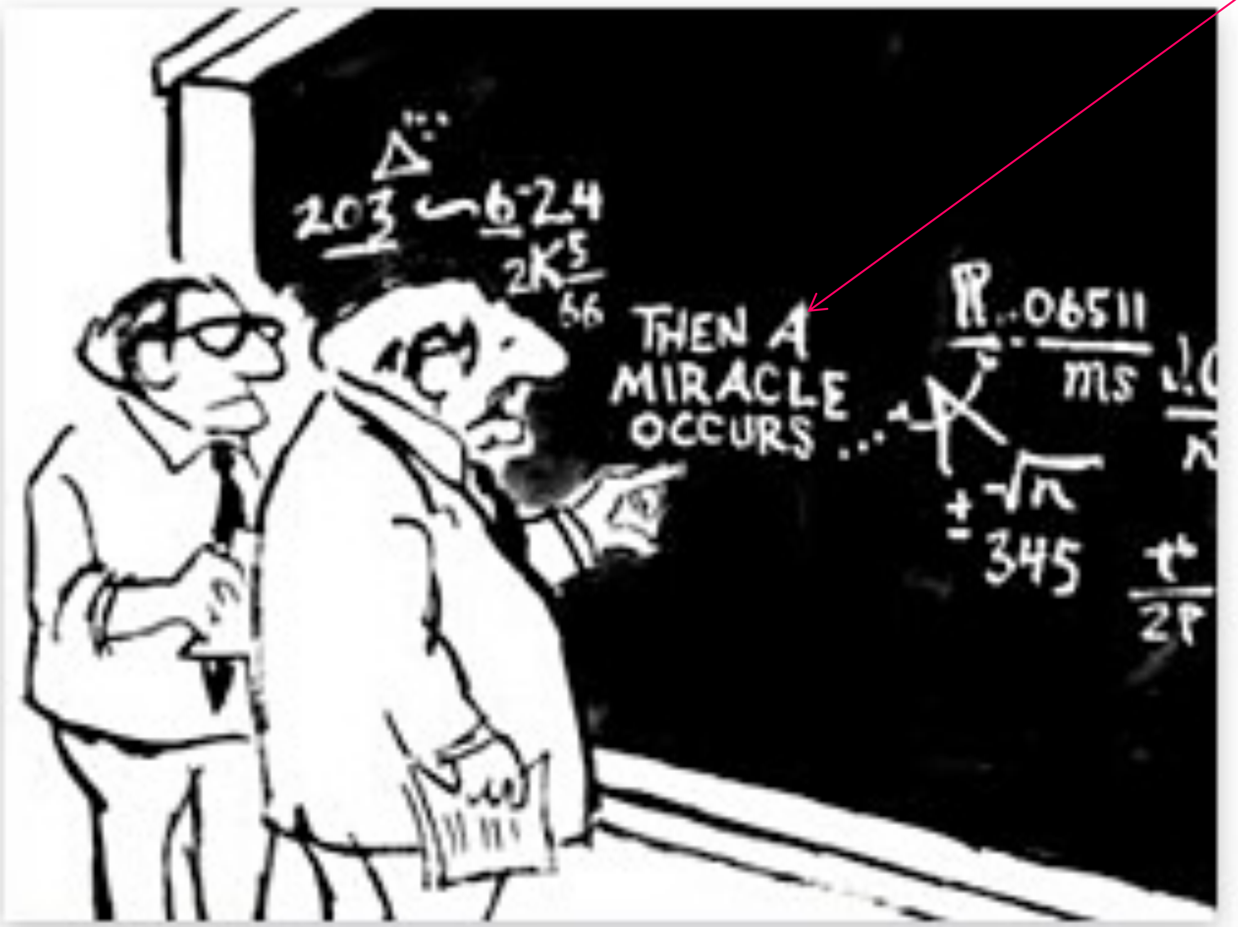
- [Nash 1950]: Every finite game has a Nash equilibrium.
- [Brouwer 1911]: Every continuous function $f: D \to D$ from a convex compact set D to itself has a fixed point $x^* = f(x^*)$.
- [Sperner 1928]: Every legal 3-coloring of a 2-d triangulated square has a tri-chromatic triangle.

- So Sperner is the culprit.

- What's happening in the proof of Sperner?

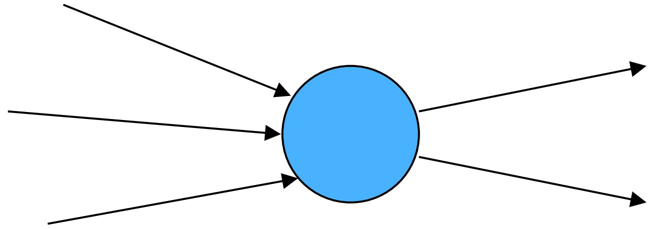
What step in Sperner's proof cannot be executed in polynomial-time?

??



An easy parity lemma:

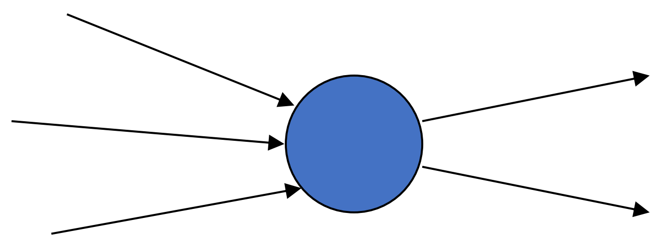
A directed graph with an unbalanced node (a node with indegree ≠ outdegree) must have another.



The Non-Constructive Step

An easy parity lemma:

A directed graph with an unbalanced node (a node with indegree ≠ outdegree) must have another.



But, wait, why is this non-constructive?

Given a directed graph and an unbalanced node, isn't it trivial to find another unbalanced node?

In some cases, the graph can be exponentially large in its succinct description…

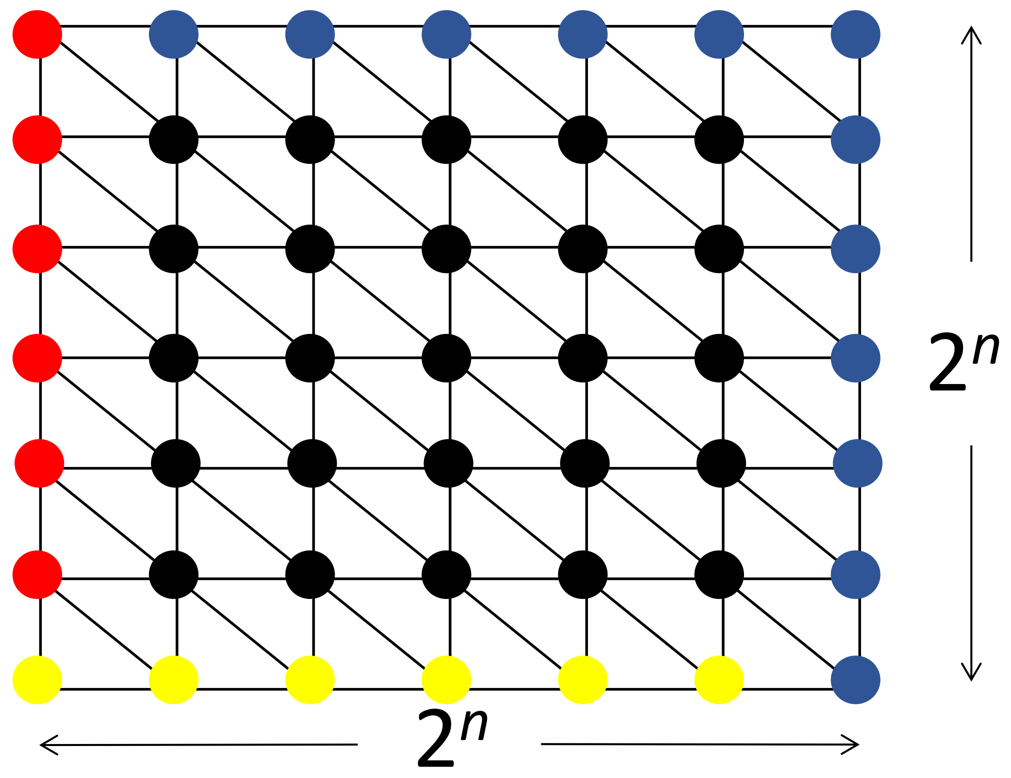
Example: next slide!

Computational Problem: SPERNER

INPUT:

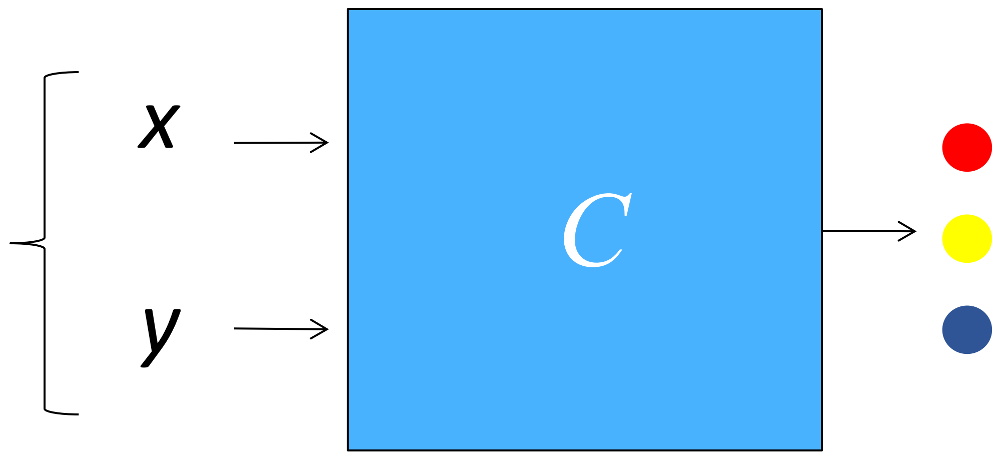
(i) *n*: specifies the size of a grid

(grid never written down!)



(ii) Imagine boundary has standard coloring shown above, while colors of internal vertices are given by a circuit:

input: the coordinates of a point *(n bits each)*



OUTPUT: A tri-chromatic triangle

exists because boundary coloring satisfies Sperner lemma constraints

but doing walk through grid to find one may take exponential time in n

Menu

Refresher: Nash, Sperner, Brouwer

Total Search Problems in NP

PPAD

NASH

INPUT: (i) A Game defined by

- the number of players n;
- an enumeration of the strategy set S_p of every player $p = 1, \ldots, n$;
- the utility function $u_p : \times_q S_q \to \mathbb{R}$ of every player.

(ii) An approximation requirement ε

OUTPUT: An ε-Nash equilibrium of the game,

$$i.e.\ x_1, \ldots, x_n \text{ s.t. } \forall i, x_i' : u_i(x_i, x_{-i}) \geq u_i(x_i', x_{-i}) - \varepsilon$$

* Approximation: Already in 1951, Nash provides a 3-player game whose unique equilibrium is irrational. This motivates our definition of the problem in terms of approximation.

** 2-player Games: 2-player games always have a rational equilibrium of polynomial description complexity in the size of the game. So we can also define the exact NASH problem for 2-player games.

BROUWER

INPUT:

(i) Turing Machine M which:

- takes n real inputs $x_1, \dots, x_n \in [0,1]$
- outputs n real outputs $y_1, \dots, y_n \in [0,1]$
- runs in polynomial time*

(ii) A constant $L > 0$ and a constant $\varepsilon > 0$

OUTPUT: Output :

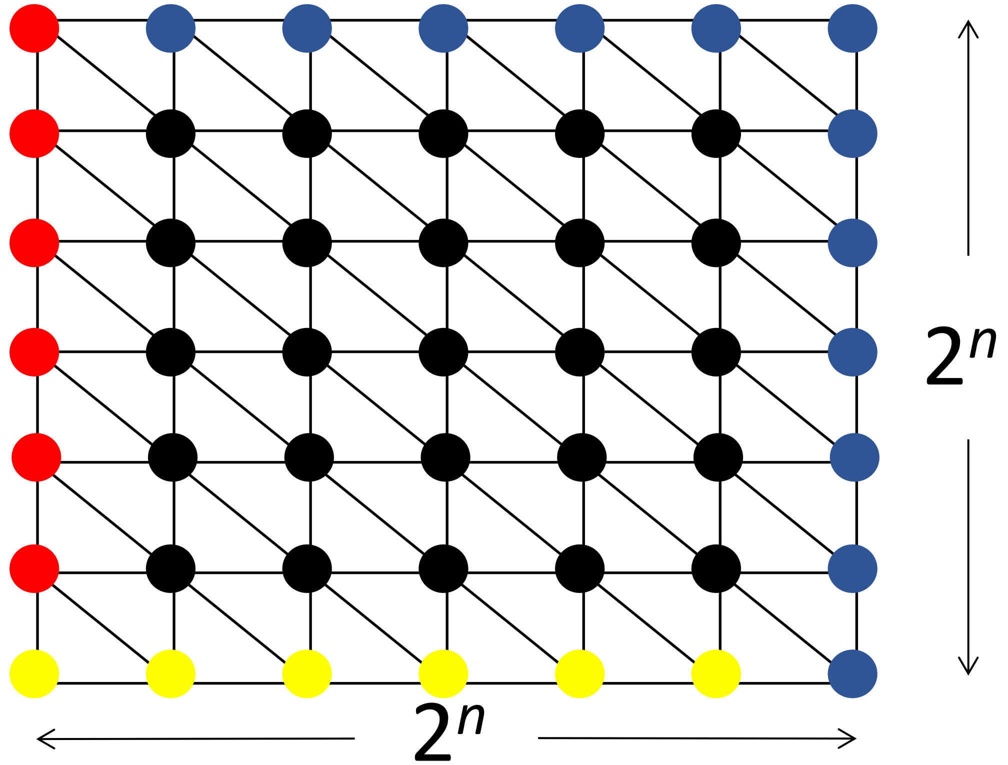
- either some fixed point x of M, i.e. satisfying $\left|\left|x - M(x)\right|\right| \leq \varepsilon$
- or some violation that $M(\cdot)$ is L-Lipschitz, i.e. a pair x, x' s.t. $||M(x) - M(x')|| > L \cdot ||x - x'||$

Computational Problem: SPERNER

INPUT:

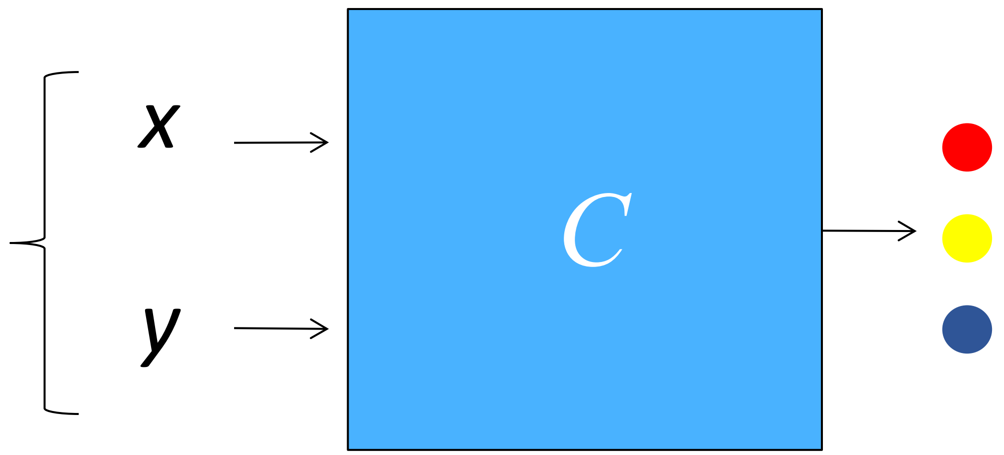
(i) n: specifies the size of a grid

(grid never written down!)



(ii) Imagine boundary has standard coloring shown above, while colors of internal vertices are given by a circuit:

input: the coordinates of a point *(n bits each)*



OUTPUT: A tri-chromatic triangle

exists because boundary coloring satisfies Sperner lemma constraints

but doing walk through grid to find one may take exponential time in n

Complexity of NASH, SPERNER, BROUWER

- No polynomial-time algorithm has been found in ~70, ~90, and ~110 years respectively
- Are they NP-hard?

- It is known that related problems *are* NP-hard:
  - e.g. determining whether a game has a Nash equilibrium with some property is NP-hard [Gilboa, Zemel ’89]
  - e.g. determining whether a game has more than one Nash equilibrium is NP-hard [Conitzer, Sandholm ’03]

- But how about the real NASH, SPERNER and BROUWER ?

Function NP (FNP)

A *search problem* L is defined by a relation $R_L \subseteq \{0,1\}^* \times \{0,1\}^*$ such that $(x, y) \in R_L$ iff y is a solution to x

A search problem is called *total* iff $\forall x. \exists\ y$ such that $(x, y) \in R_L$.

A search problem $L \in$ FNP iff there exists a poly-time algorithm $A_L(\cdot, \cdot)$ and a polynomial function $p_L(\cdot)$ such that

(i) $\forall\, x, y$: $\quad A_L(x, y)=1 \Leftrightarrow (x, y) \in R_L$

(i.e. a solution y to instance x can be verified as such in polynomial time)

(ii) $\forall\, x$: $\exists\, y$ s.t. $(x, y) \in R_L \Rightarrow \exists\, z$ with $|z| \leq p_L(|x|)$ s.t. $(x, z) \in R_L$

(i.e. whenever x has solution, x has solution of polynomial length)

TFNP = $\{L \in \text{FNP} \mid L \text{ is total}\}$

SPERNER, NASH, BROUWER $\in$ FNP.

FNP-completeness

A search problem $L \in$ FNP, associated with A_L and p_L, is ***poly-time (Karp) reducible*** to another problem $L' \in$ FNP, associated with $A_{L'}$ and $p_{L'}$, iff there exist efficiently computable functions f, g such that

(i) $f : \{0,1\}^* \rightarrow \{0,1\}^*$ maps inputs x to L into inputs $f(x)$ to L'

(ii) $\forall\, x, y$: y solves $f(x)$ $\Rightarrow$ $g(y)$ solves x

$\forall$ x: $f(x)$ has no solution $\Rightarrow$ x has no solution

can't Karp reduce SAT to SPERNER, NASH or BROUWER

A search problem L is ***FNP-complete*** iff

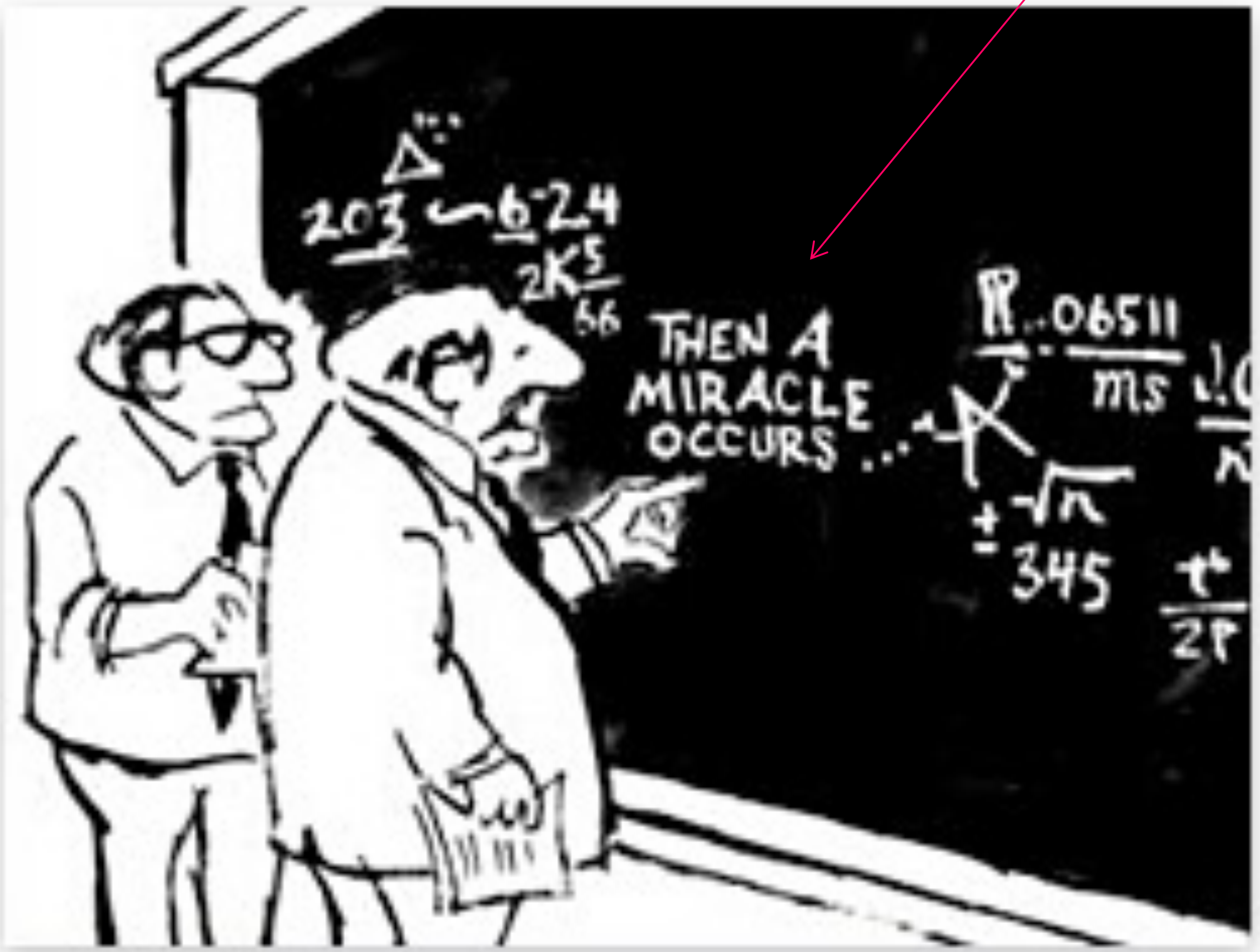
e.g. SAT

$L \in$ FNP

L' is poly-time reducible to L, for all $L' \in$ FNP

A Complexity Theory of Total Search Problems ?

??



A Complexity Theory of Total Search Problems ?

100-feet overview of our methodology:

1. identify the combinatorial argument of existence, responsible for making these problems total;

2. define a complexity class inspired by the argument of existence;

3. make sure that the complexity of the problem was captured as tightly as possible (via completeness results).

Menu

Refresher: Nash, Sperner, Brouwer

Total Search Problems in NP

PPAD

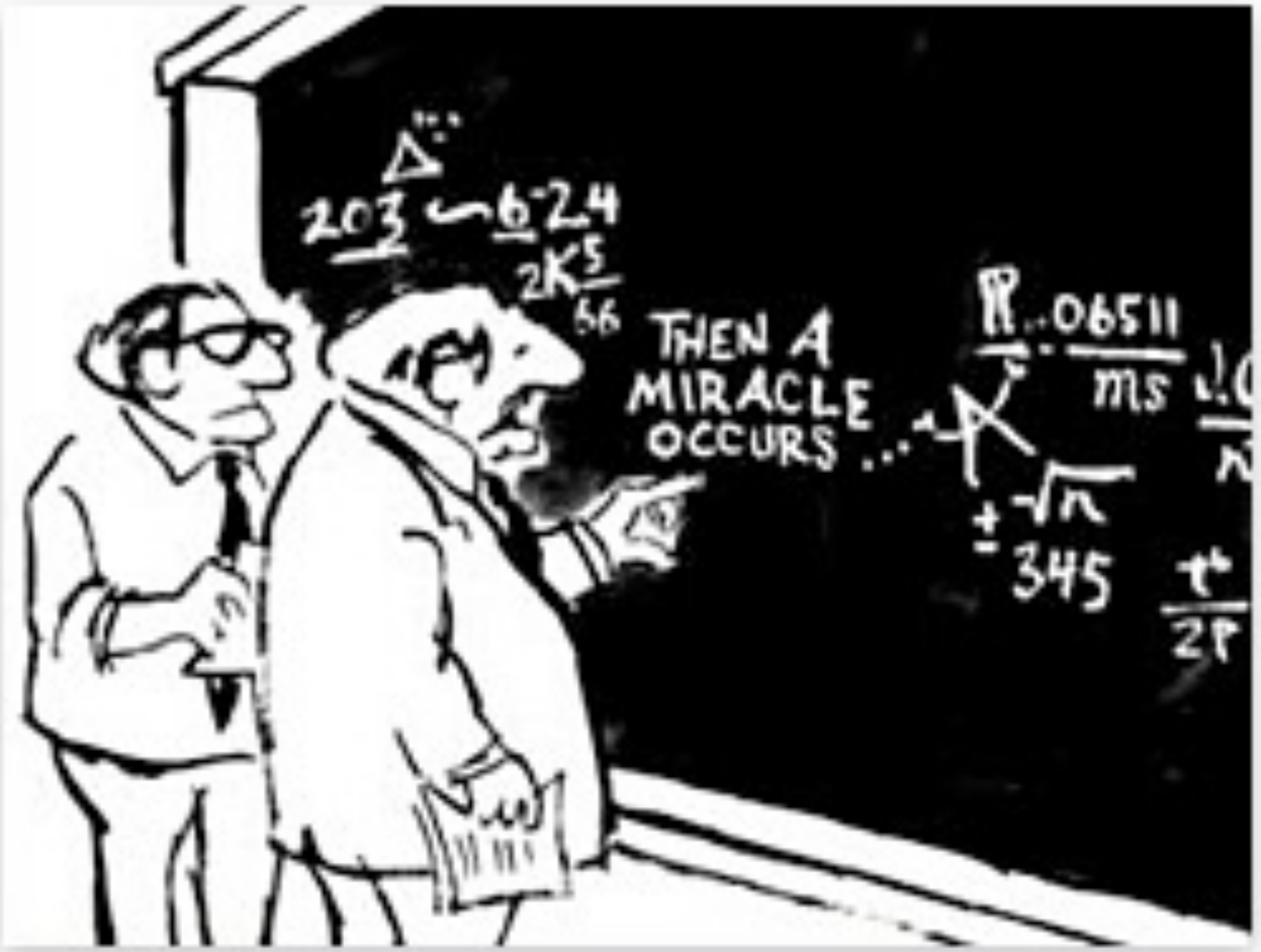
Menu

Refresher: Nash, Sperner, Brouwer

Total Search Problems in NP

PPAD

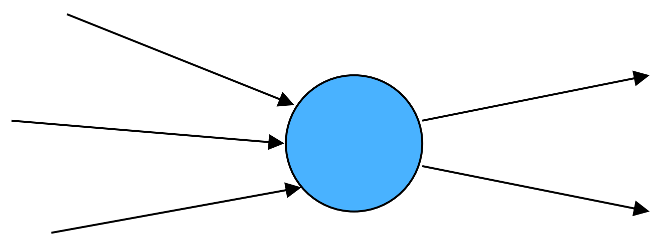
OK, so what is the combinatorial argument of existence underlying Sperner, Brouwer and Nash?



The Non-Constructive Step

An easy parity lemma:

A directed graph with an unbalanced node (a node with indegree $\neq$ outdegree) must have another.



But, wait, why is this non-constructive?

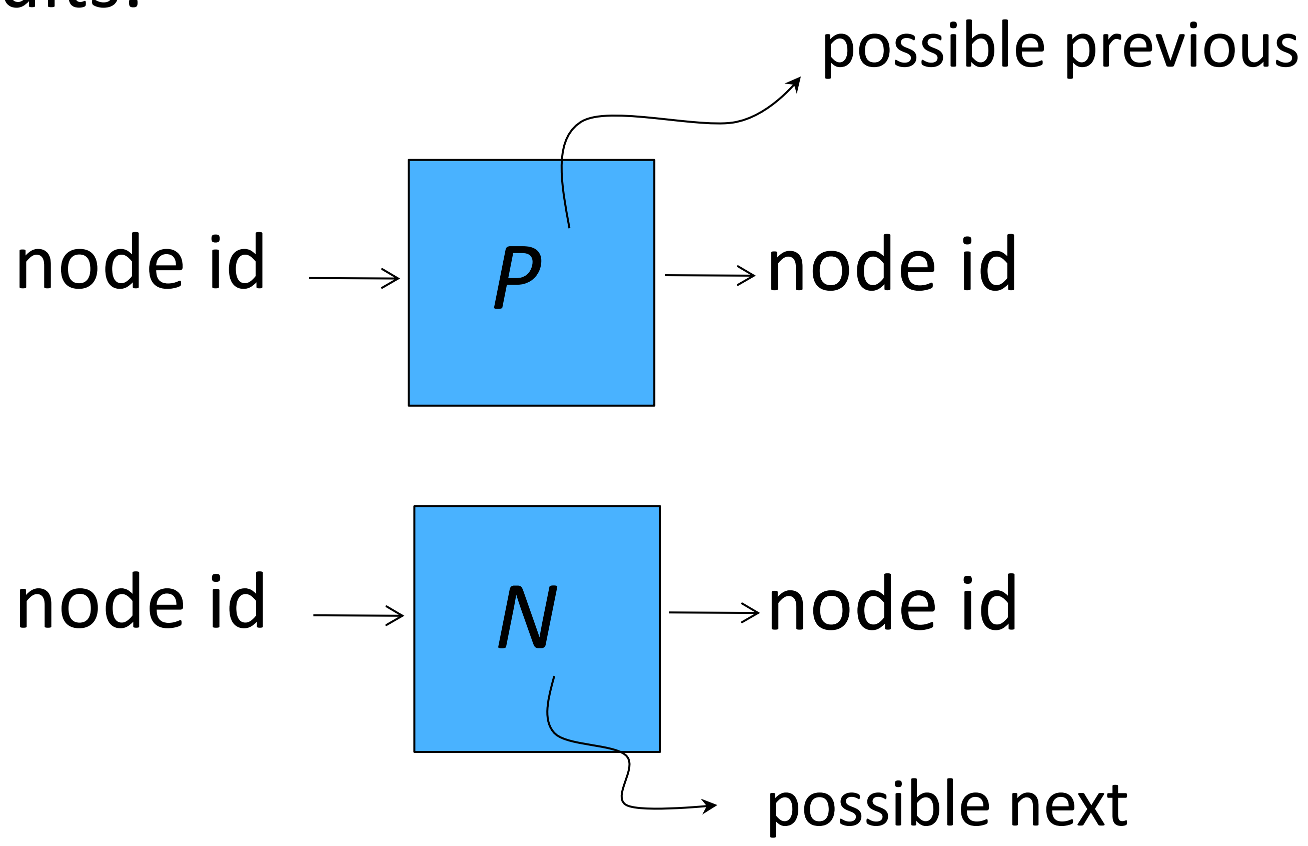
Given a directed graph and an unbalanced node, isn't it trivial to find another unbalanced node?

The graph might be exponentially larger than its description!

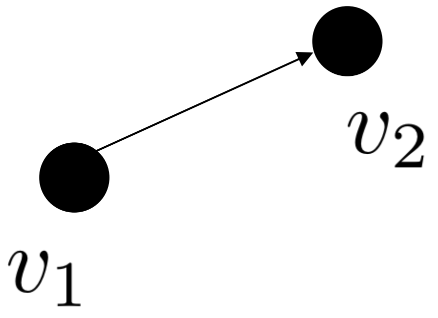
The PPAD Class [Papadimitriou ’94]

a complexity class capturing TFNP problems whose totality is due to the directed parity argument

Suppose that an exponentially large graph with vertex set $\{0,1\}^n$ is defined by two circuits:



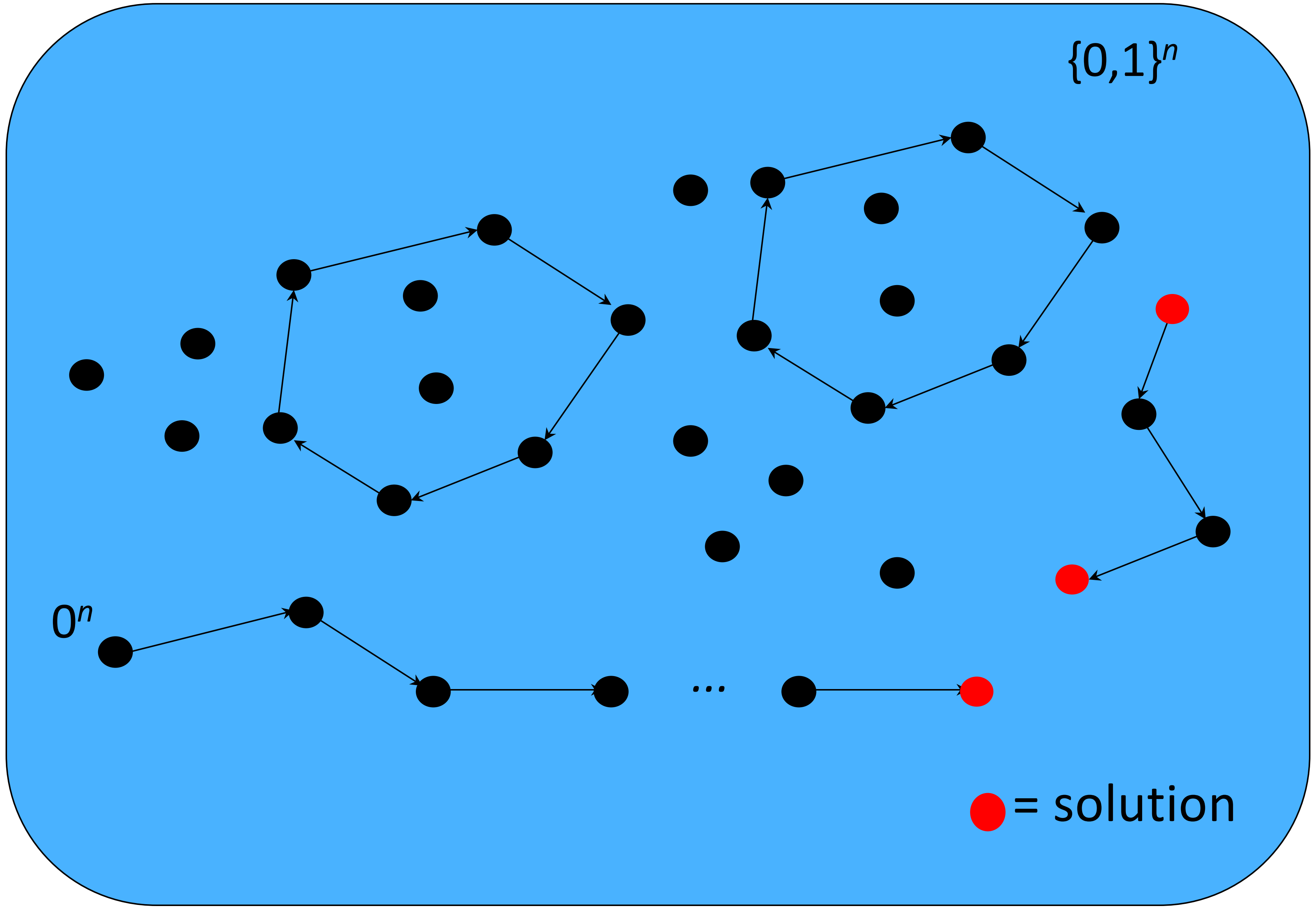
$$P(v_2) = v_1 \wedge N(v_1) = v_2$$

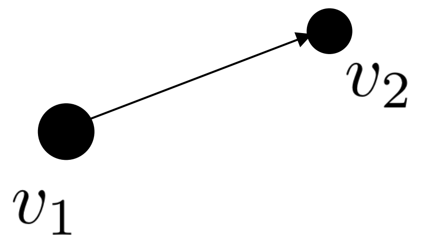


END OF THE LINE: Given P and N: If 0^n is an unbalanced node, find another unbalanced node. Otherwise output 0^n.

PPAD = { *Search problems in* FNP *reducible to* END OF THE LINE }

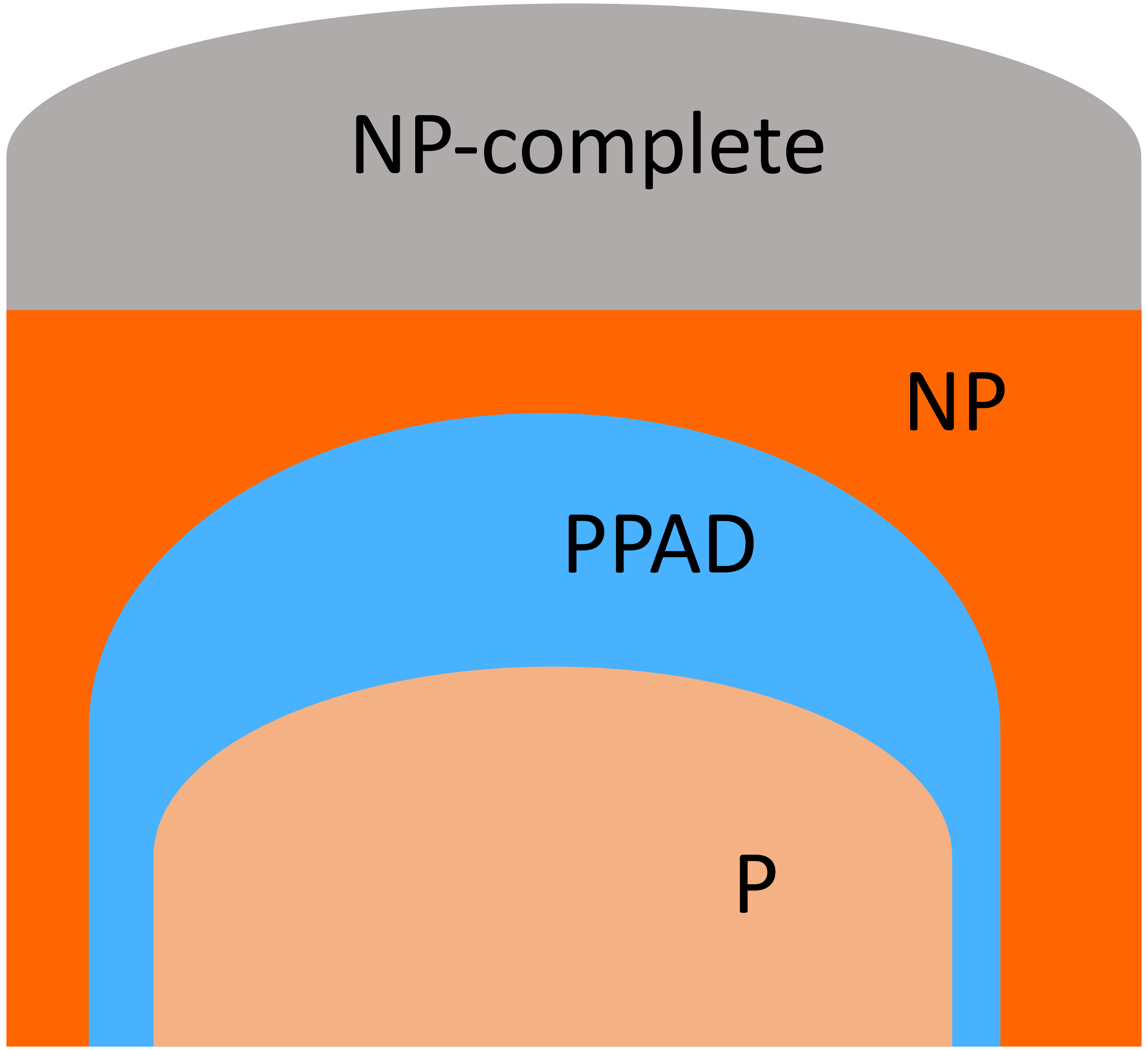
END OF THE LINE





$$\Leftrightarrow \quad P(v_2) = v_1 \wedge N(v_1) = v_2$$

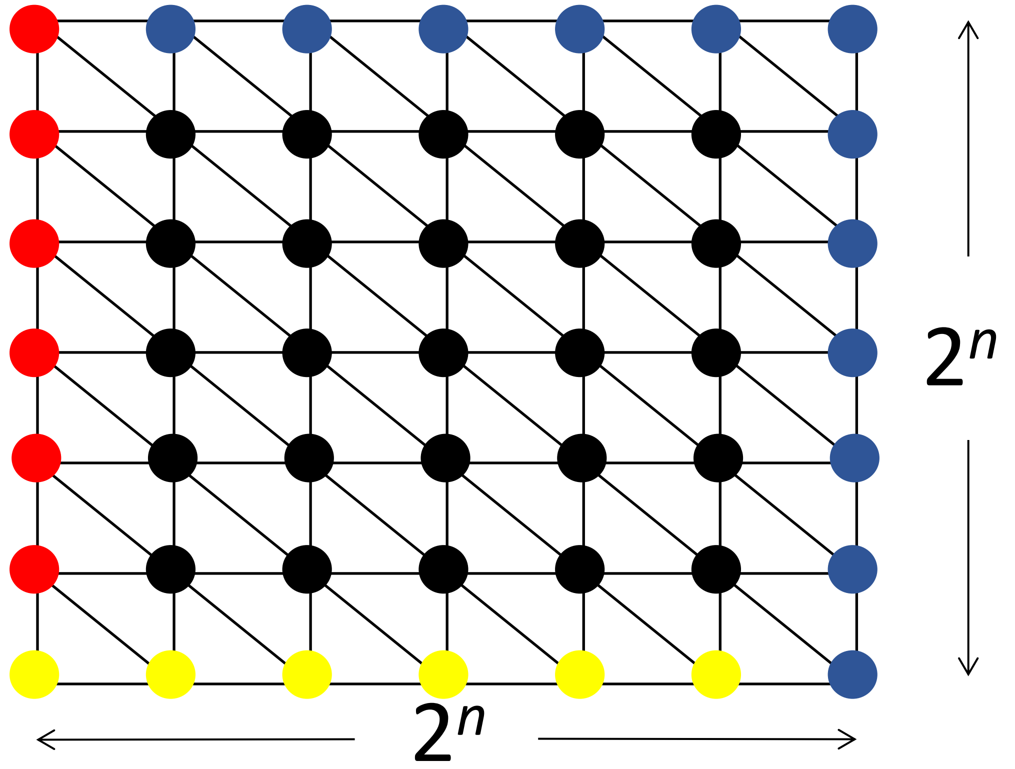
Believed Location of PPAD



e.g. in SPERNER…

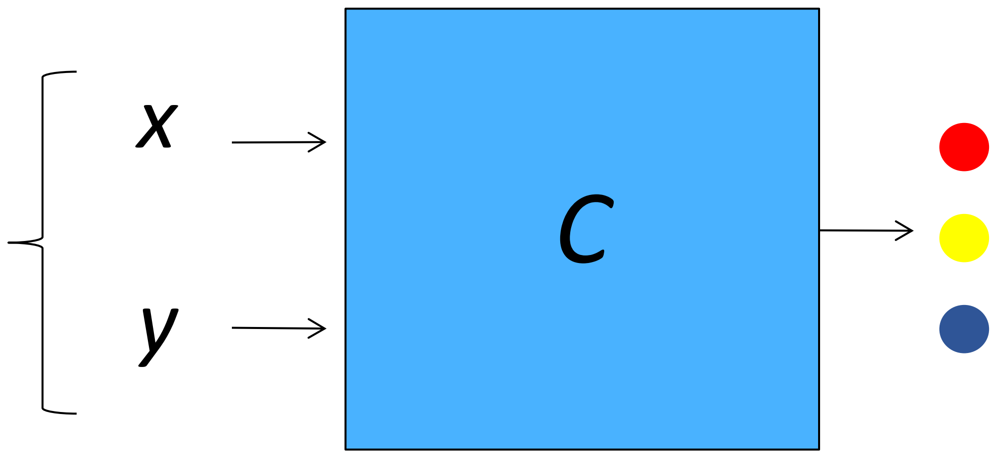
INPUT:

(i) n: specifies the size of a grid



(ii) Suppose boundary has coloring shown above, and colors of internal vertices are given by a circuit:

input: the coordinates of a point *(n bits each)*

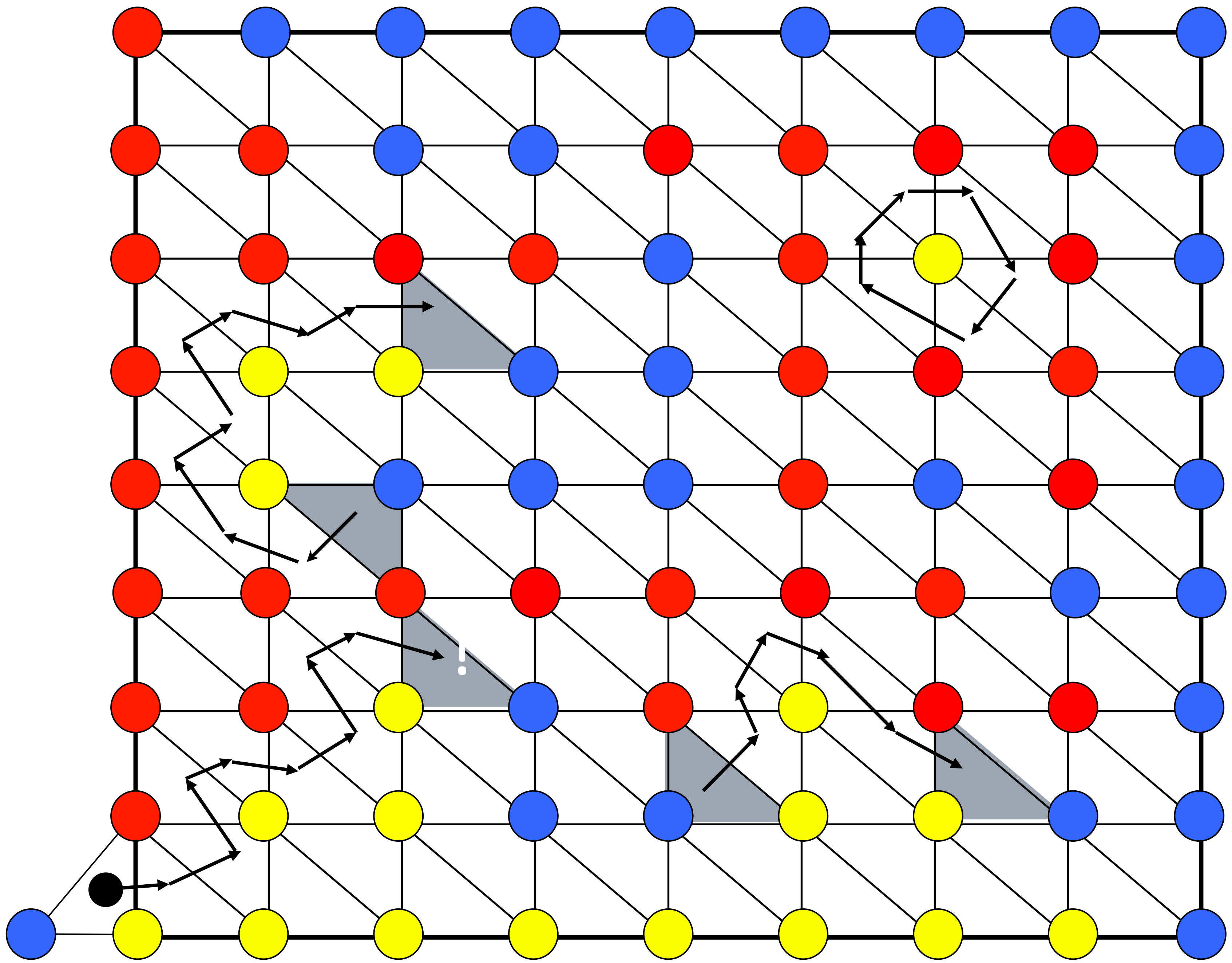


graph has size $\approx 2^{2n}$

circuit description might be $\ll 2^{2n}$

OUTPUT: A tri-chromatic triangle (exists because boundary coloring satisfies Sperner lemma constraints)

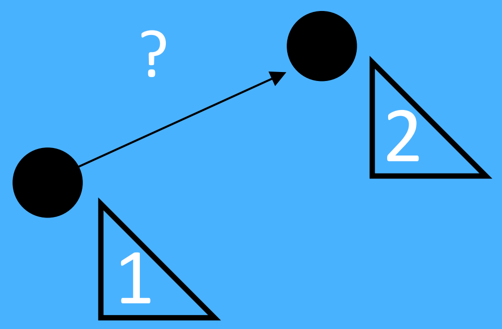
Proof of Sperner's Lemma



[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.

Proof of Sperner's Lemma

Transition Rule:

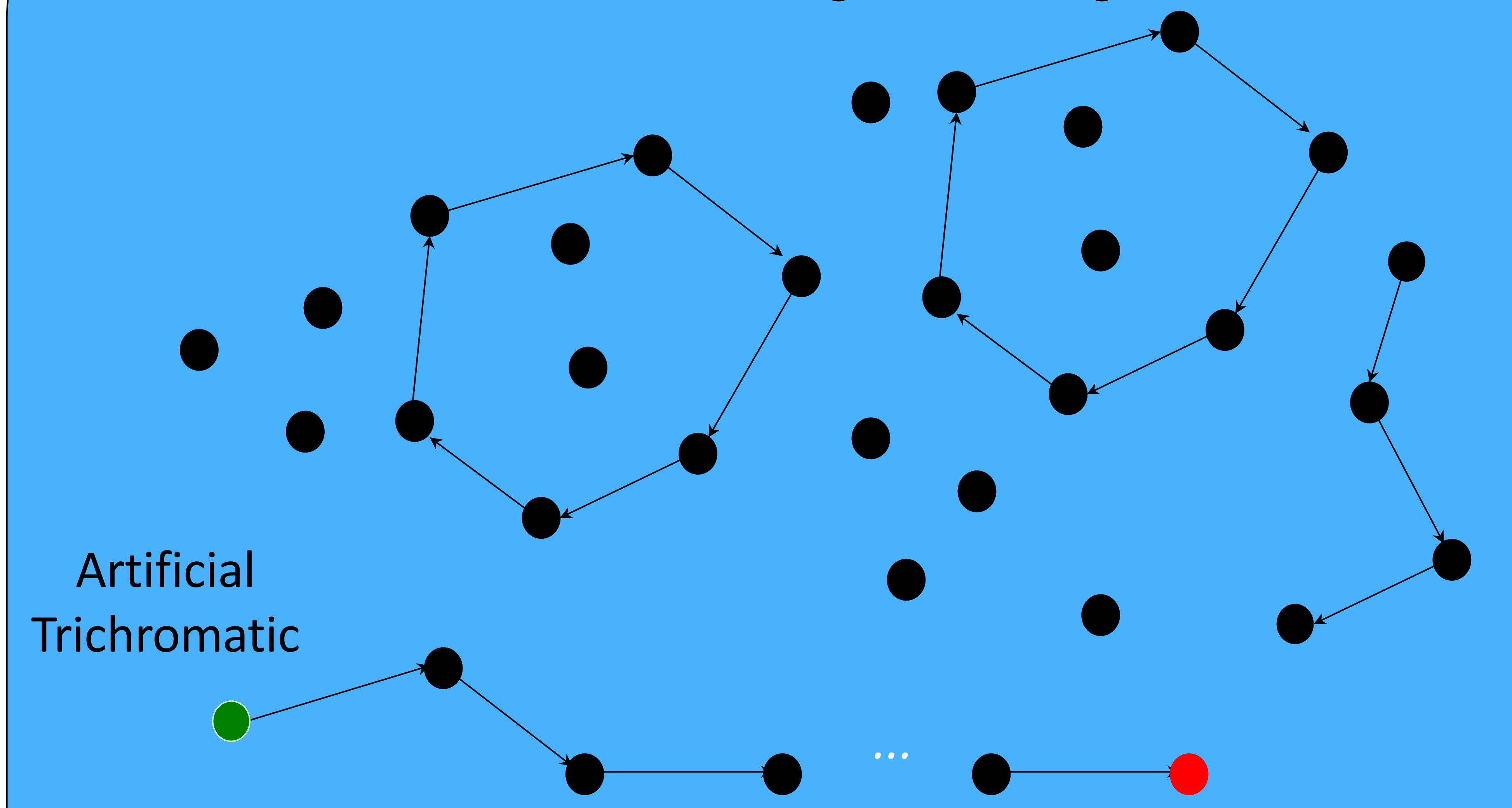


Put an edge from triangle 1 to triangle 2 if they share a ***red*** *-* ***yellow*** *edge that can be crossed to go from triangle 1 to triangle 2 keeping* ***red*** *on the left hand.*

[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.

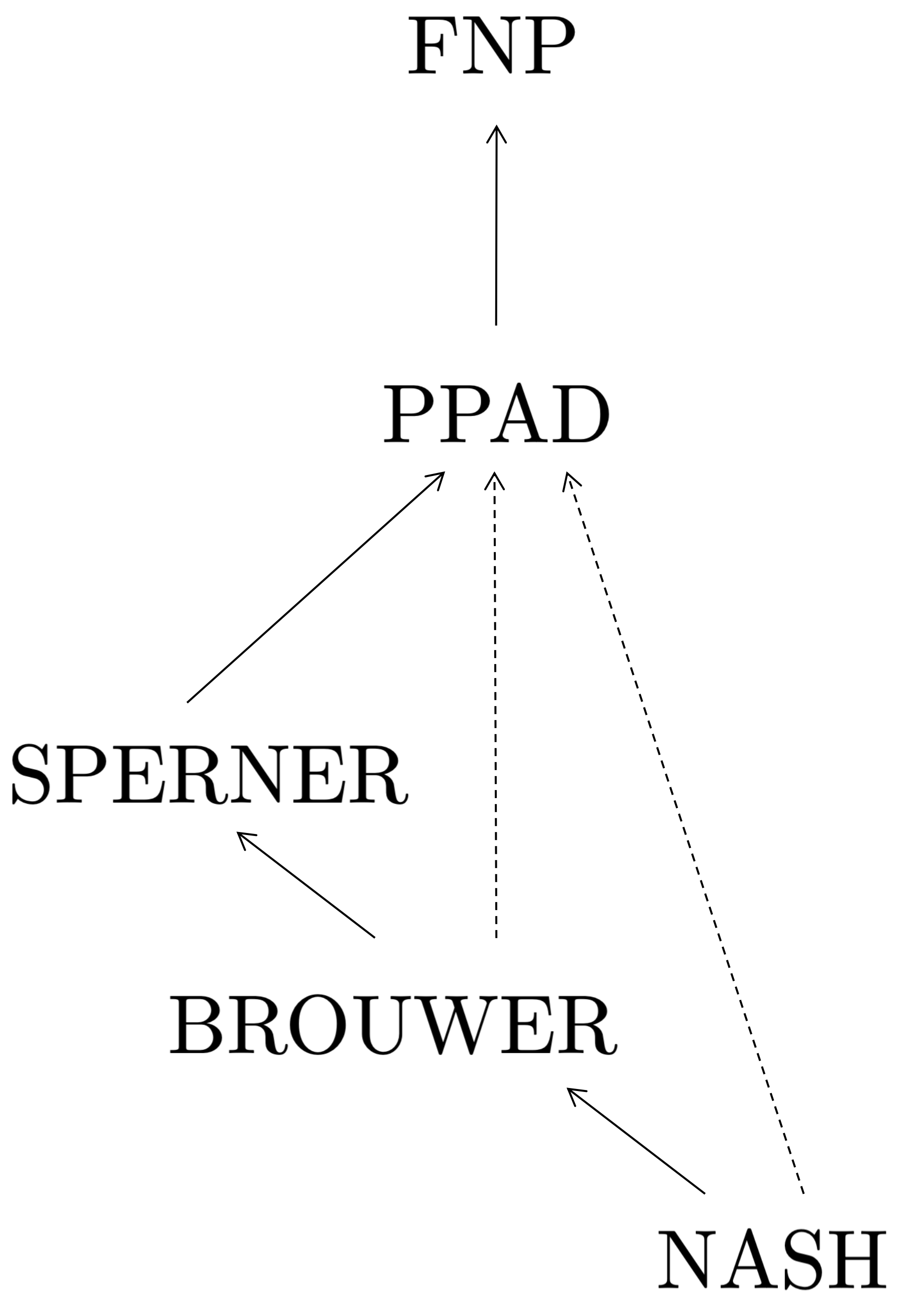
Structure of Proof:
A directed parity argument

Vertices of Graph ≡ Triangles
all vertices have in-degree, out-degree ≤ 1



degree 1 vertices: trichromatic triangles
degree 2 vertices: no blue, non-trichromatic
degree 0 vertices: all other triangles

Proof: ∃ at least one trichromatic (artificial one) ⇒ ∃ another trichromatic
Also: degree 1 vertices are in pairs but one is fake ⇒ ∃ odd number of trichromatic!



→ means poly-time reduction

Menu

Refresher: Nash, Sperner, Brouwer

Total Search Problems in NP

PPAD

Menu

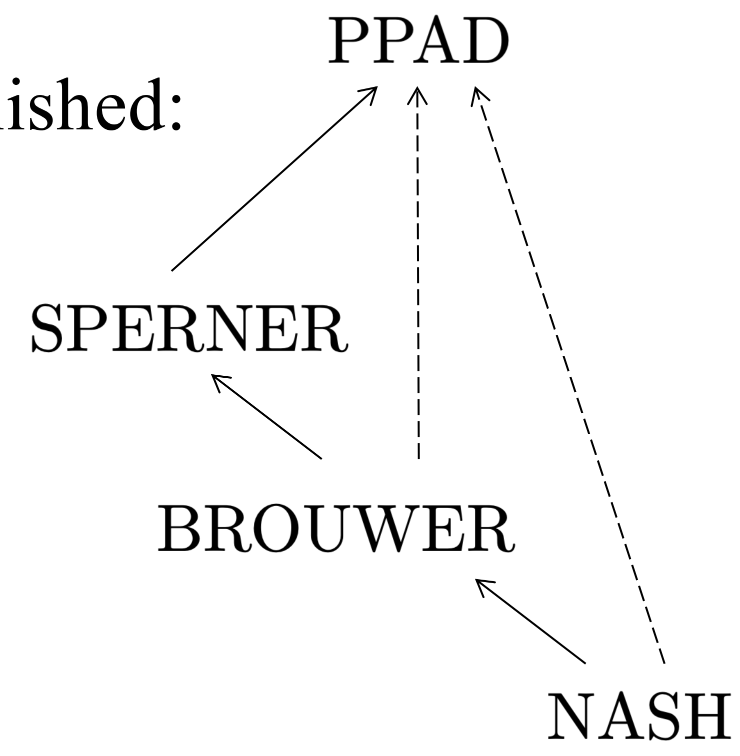
Refresher: Nash, Sperner, Brouwer

Total Search Problems in NP

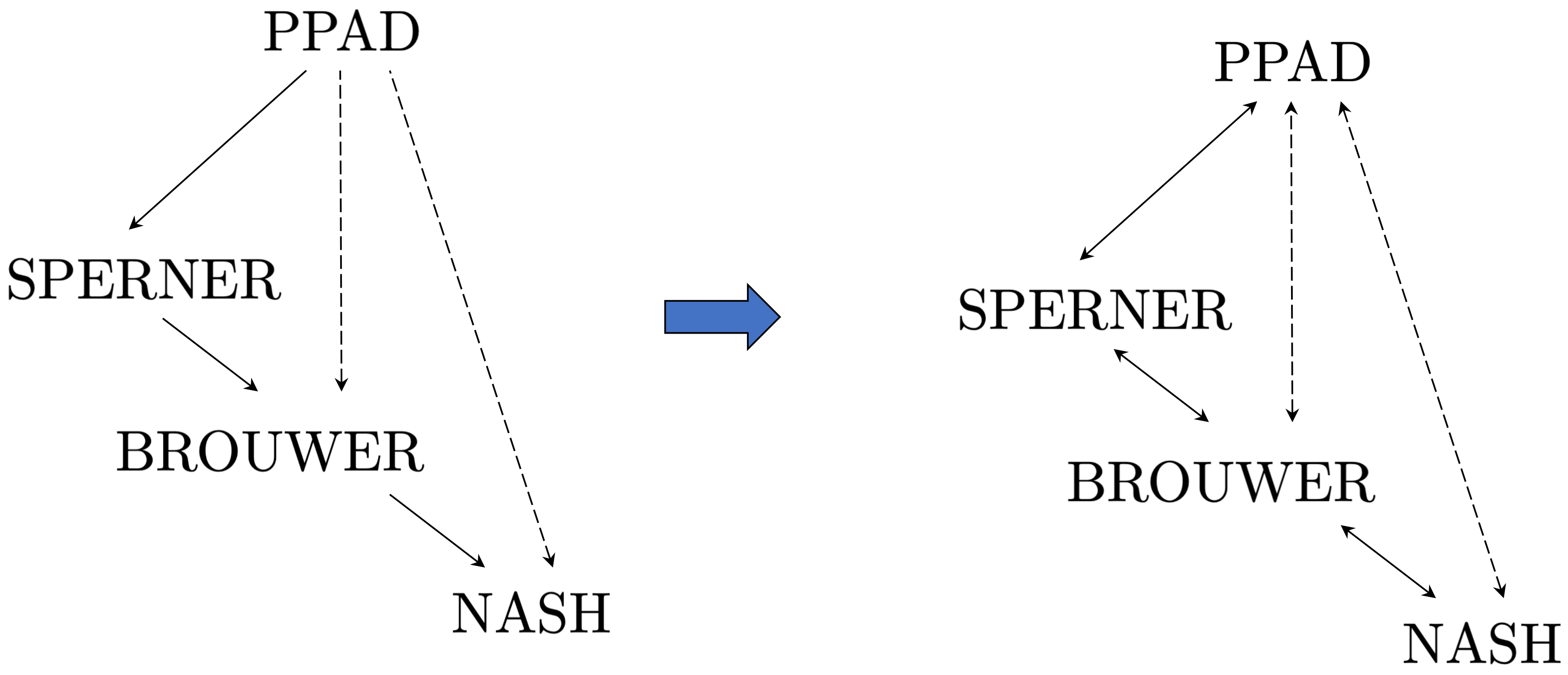
PPAD

Litmus Test: Are Nash Brouwer and Sperner PPAD-complete?

Poly-time Reductions that we just established:

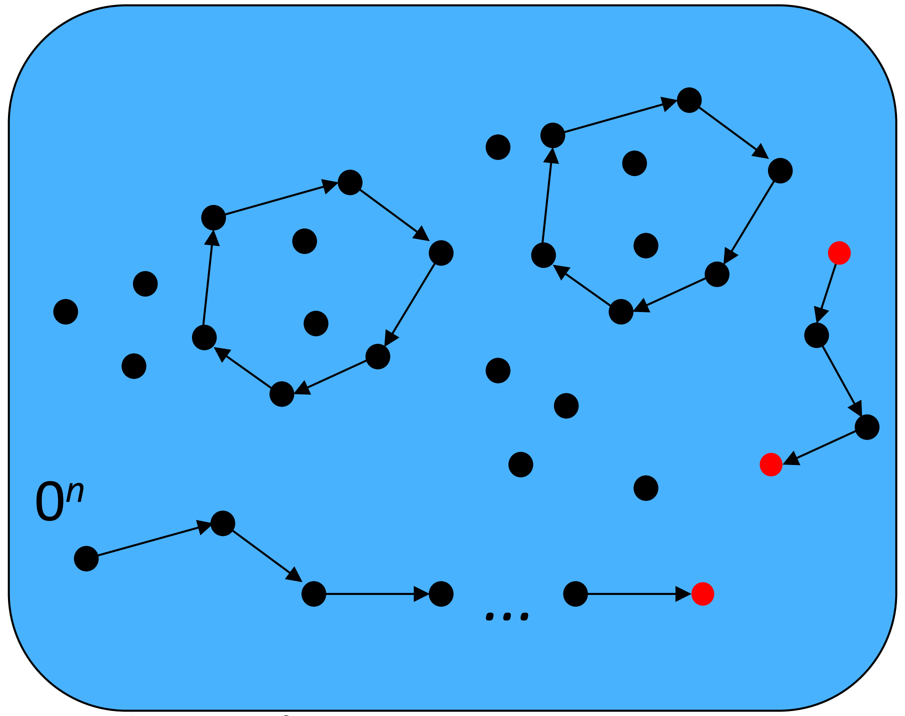


[Daskalakis-Goldberg-Papadimitriou’06]:

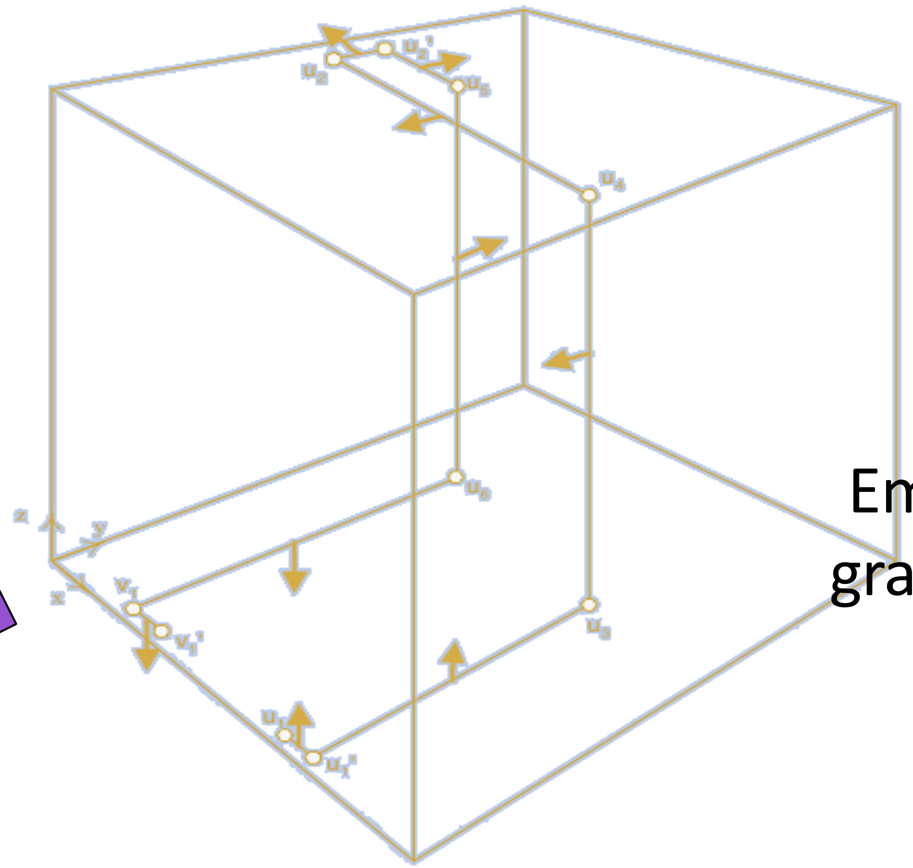


PPAD-Completeness of NASH

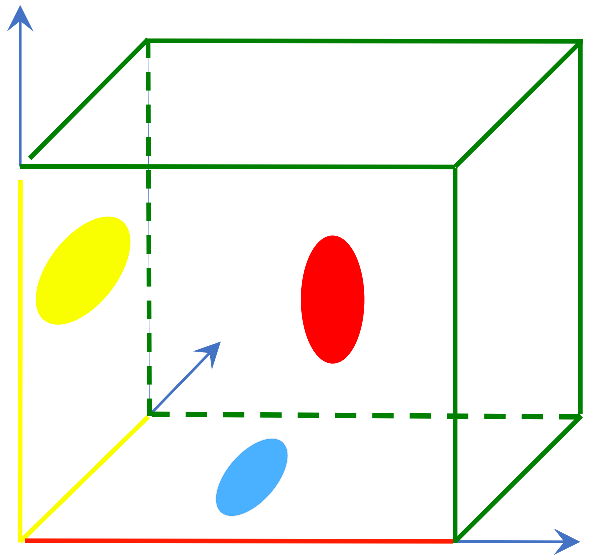
[Daskalakis, Goldberg, Papadimitriou'06]



Generic PPAD

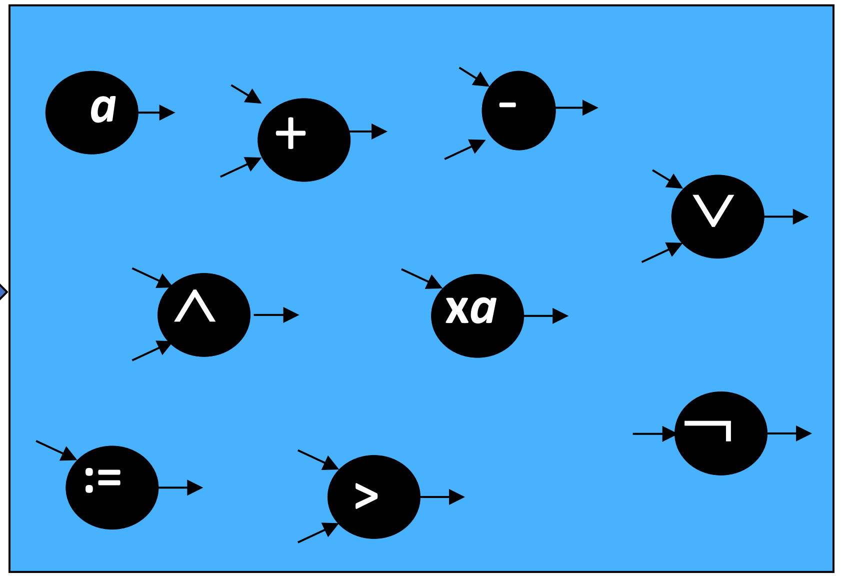


Embed PPAD graph in $[0,1]^3$

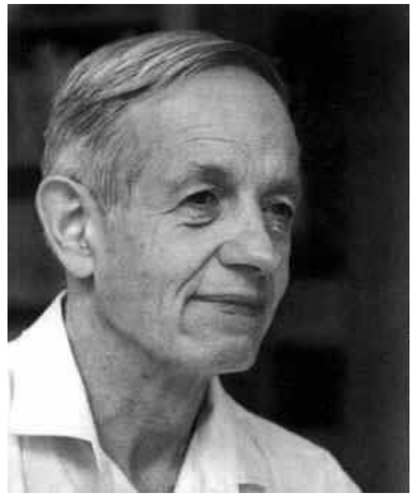


3D-SPERNER

3D-BROUWER

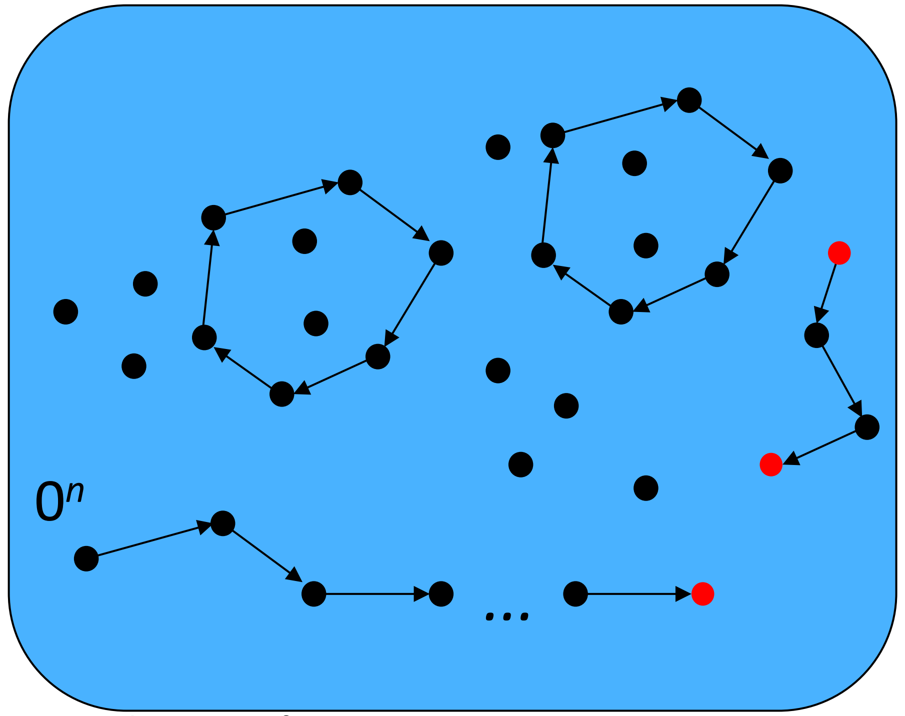


ARITHMCIRCUITSAT

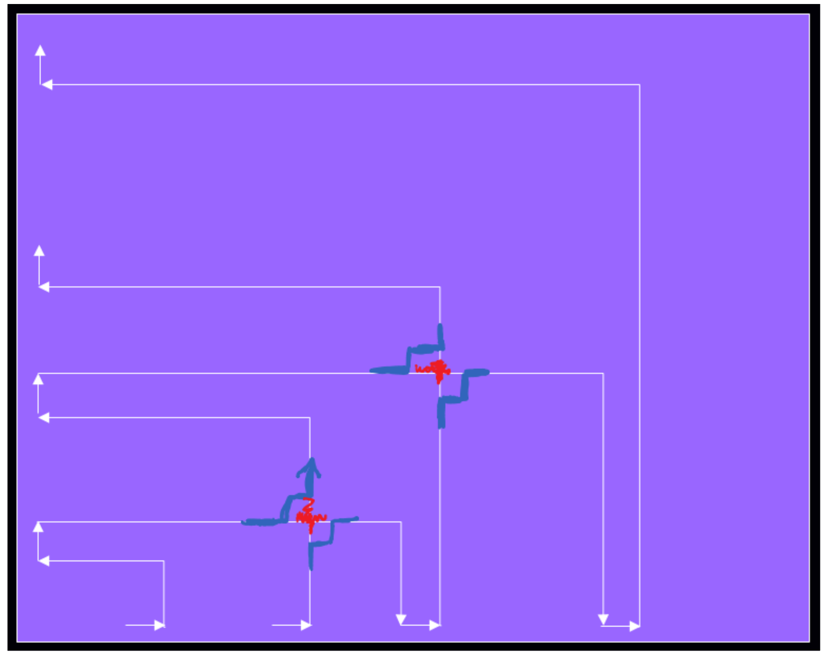


NASH

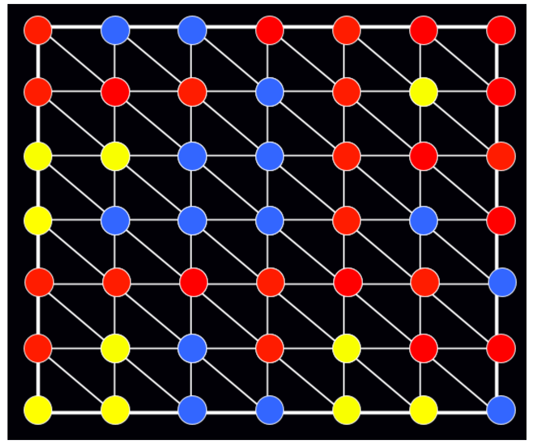
PPAD-Completeness of NASH [Daskalakis, Goldberg, Papadimitriou'06]



Generic PPAD

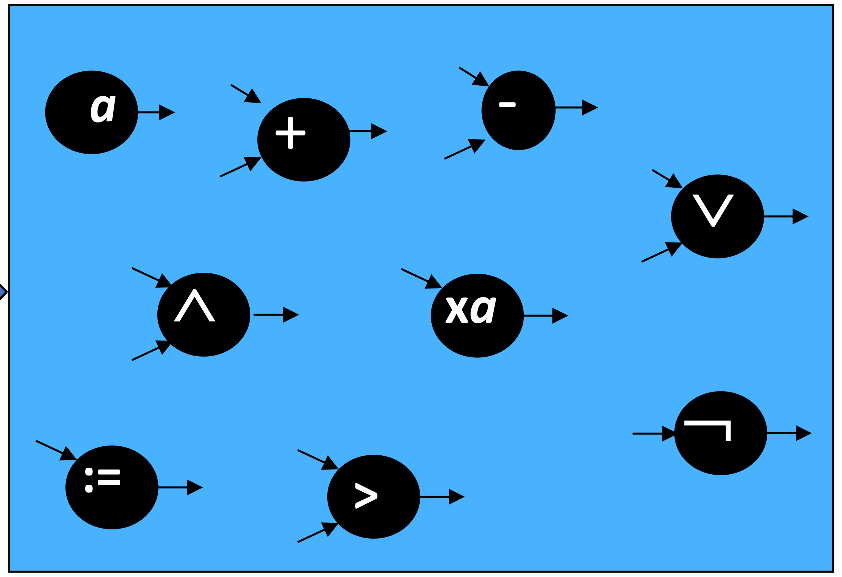


Embed PPAD graph in $[0,1]^2$

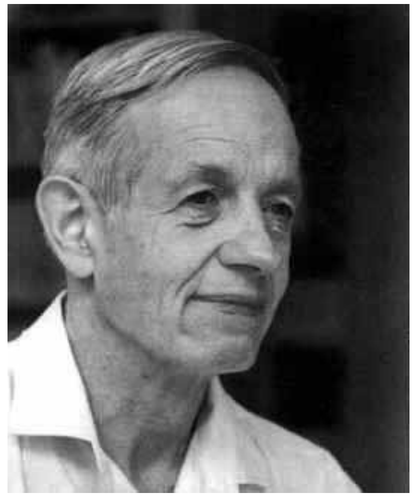


2D-SPERNER

2D-BROUWER

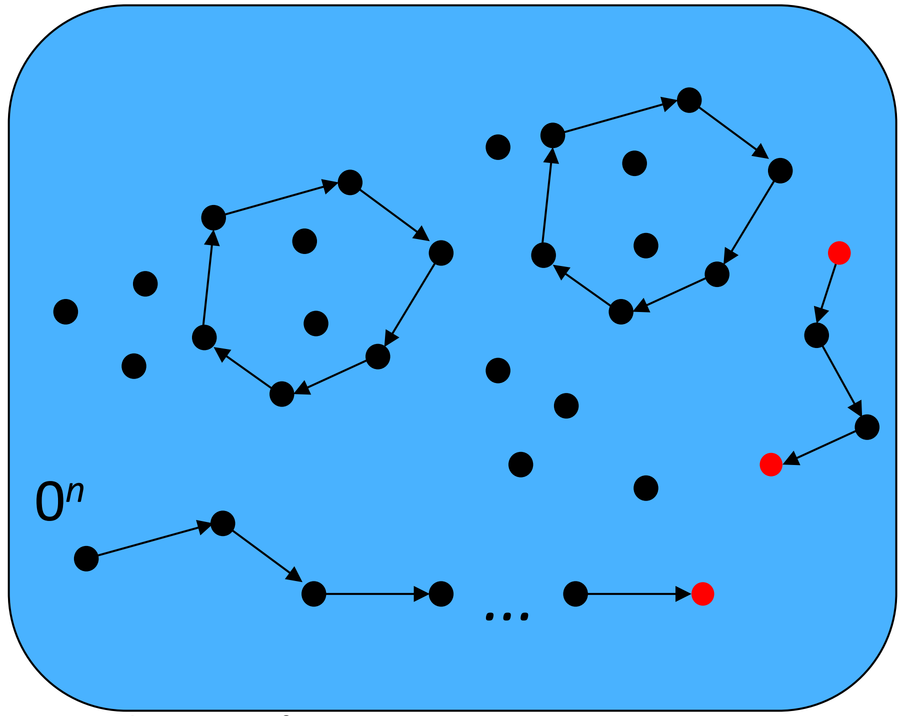


ARITHMCIRCUITSAT

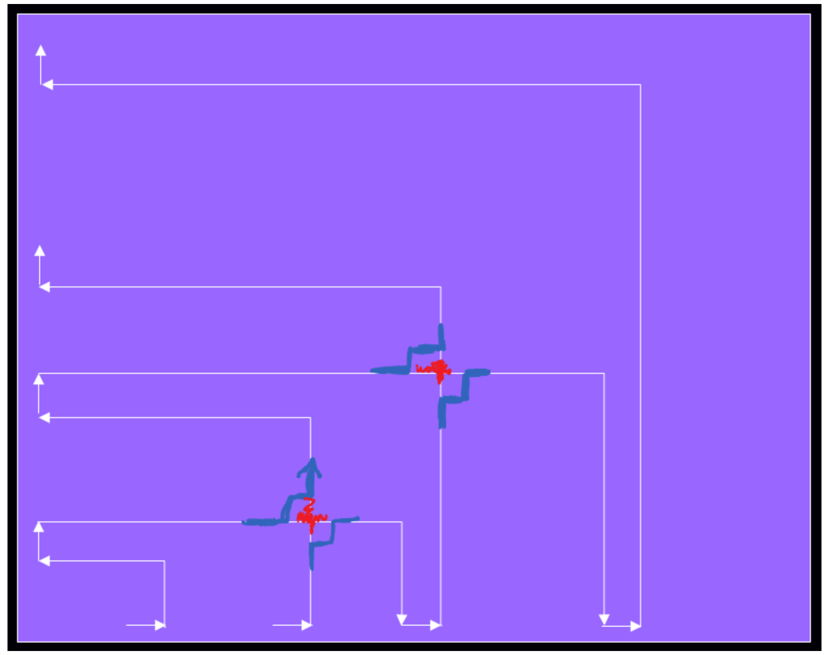


NASH

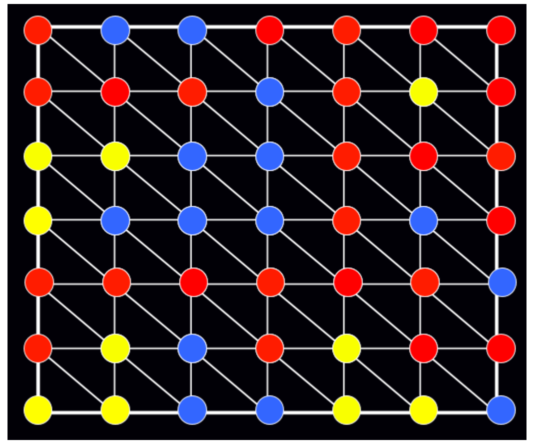
PPAD-Completeness of NASH [Daskalakis, Goldberg, Papadimitriou'06]



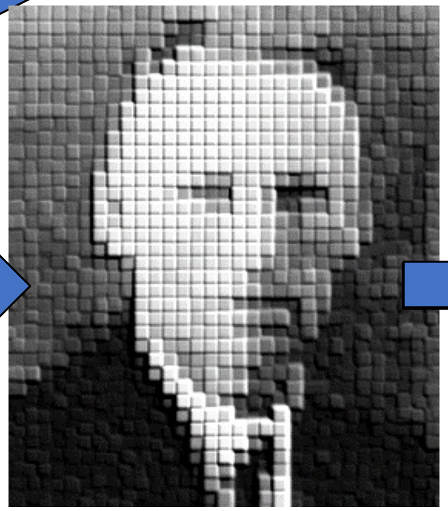
Generic PPAD



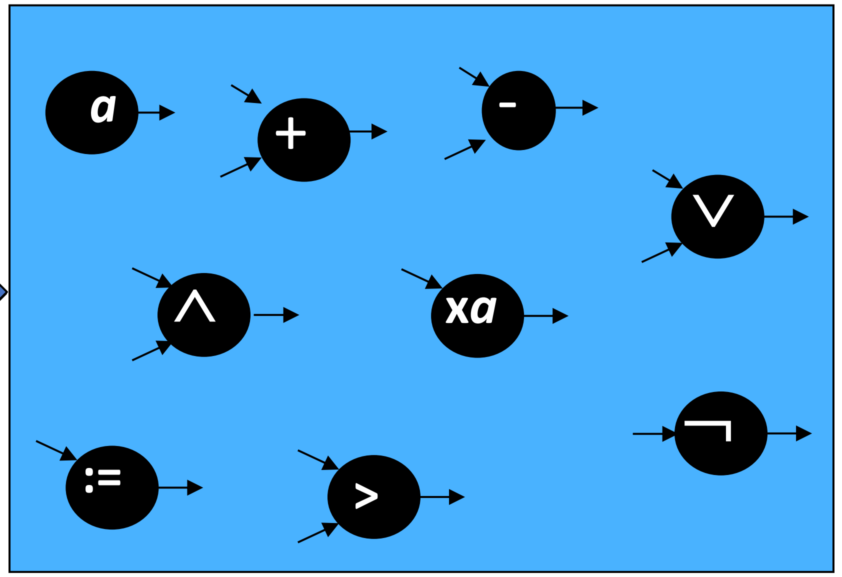
Embed PPAD graph in $[0,1]^2$



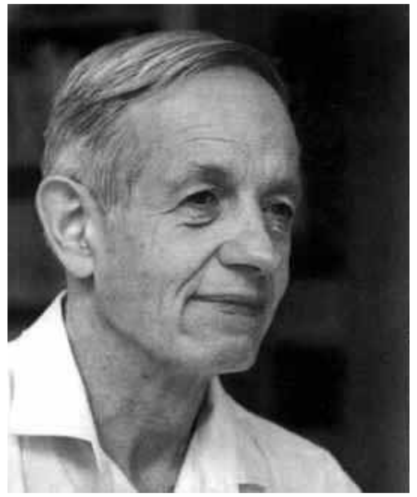
2D-SPERNER



2D-BROUWER



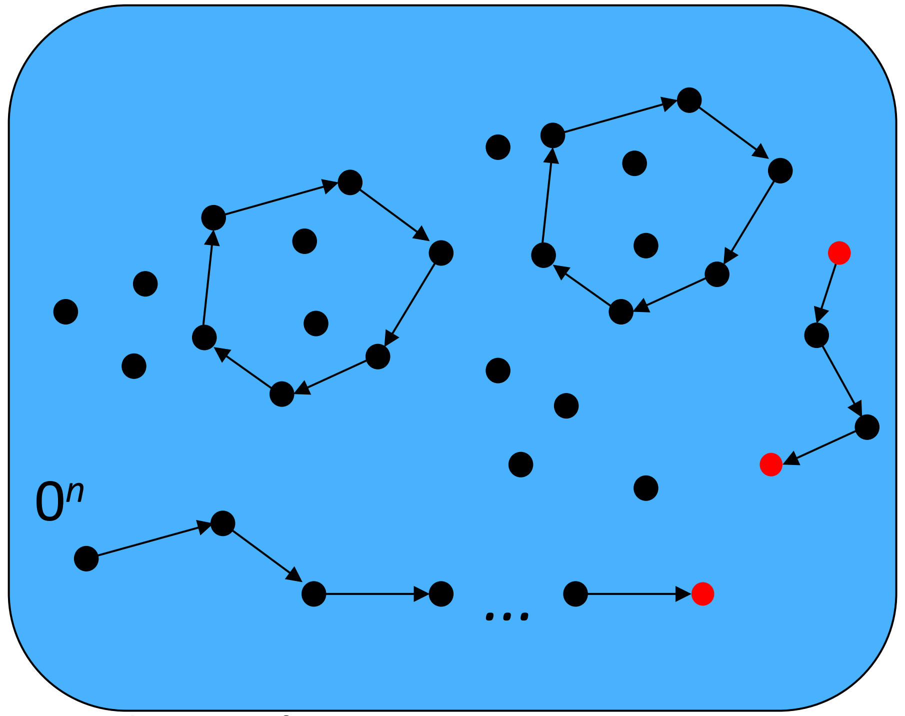
ARITHMCIRCUITSAT



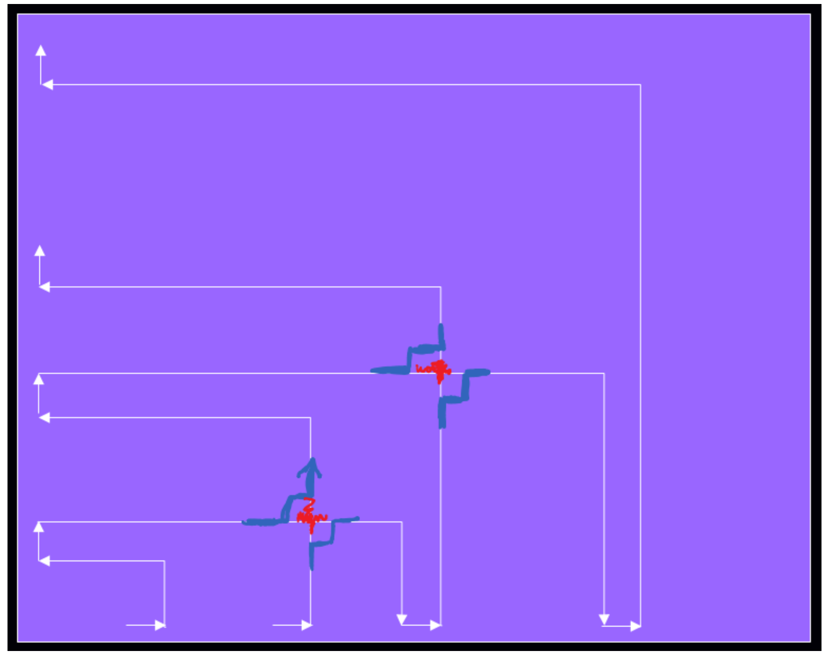
NASH

PPAD-Completeness of NASH

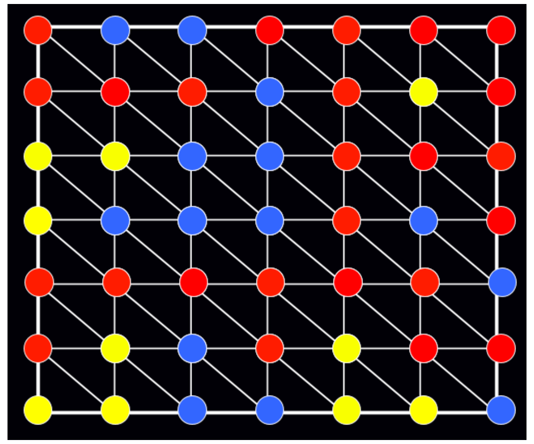
[Daskalakis, Goldberg, Papadimitriou'06]



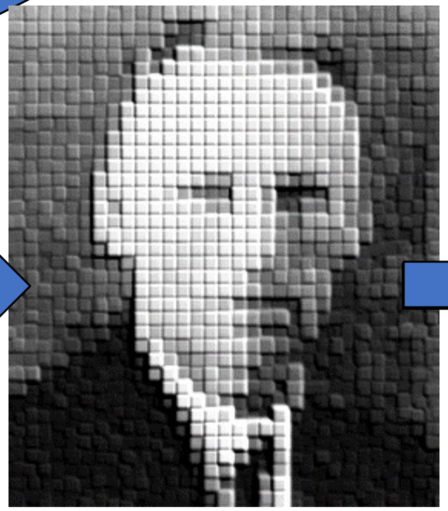
Generic PPAD



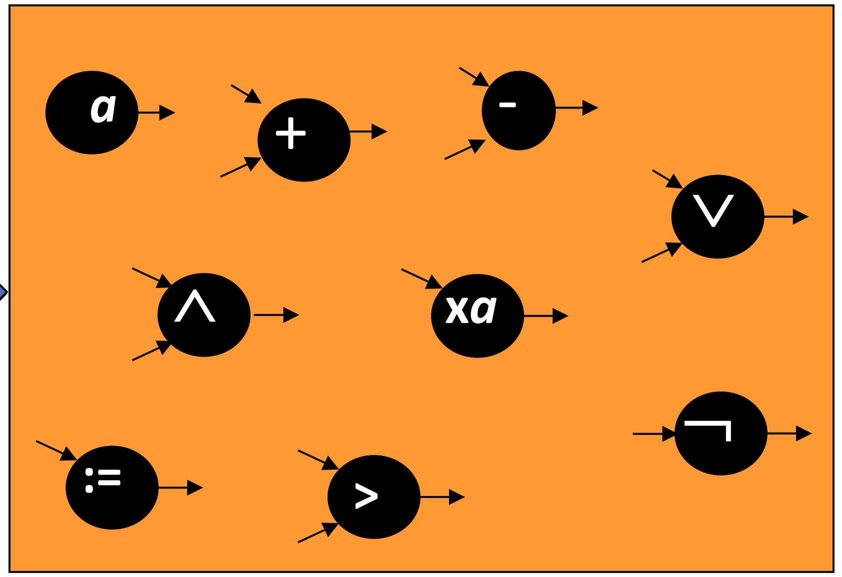
Embed PPAD graph in $[0,1]^2$



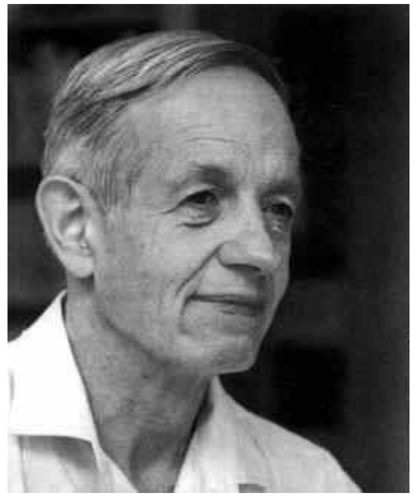
2D-SPERNER



2D-BROUWER



ARITHMCIRCUITSAT



NASH