

# The Complexity of Nash Equilibrium

[Daskalakis, Goldberg, Papadimitriou '06]: Finding a Nash equilibrium is **PPAD-complete**.

[Chen, Deng'06]: ...even in two-player games.

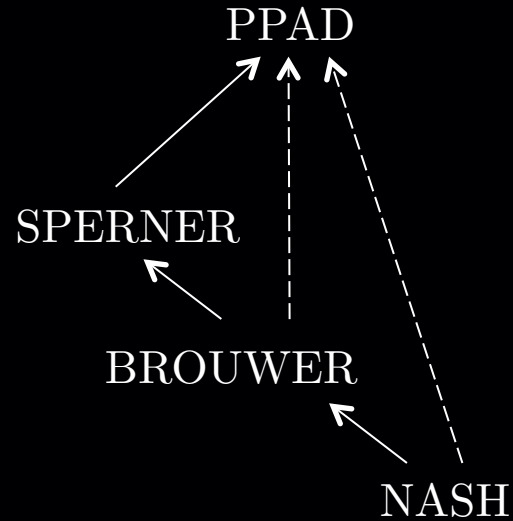
I.e. finding a Nash equilibrium is *computationally intractable*, exactly as intractable as the class **PPAD**, SPERNER, BROUWER

[Condenotti et al'06,...,Chen et al'13]: Arrow-Debreu equilibria (in markets w/ complements) are also **PPAD-hard**.

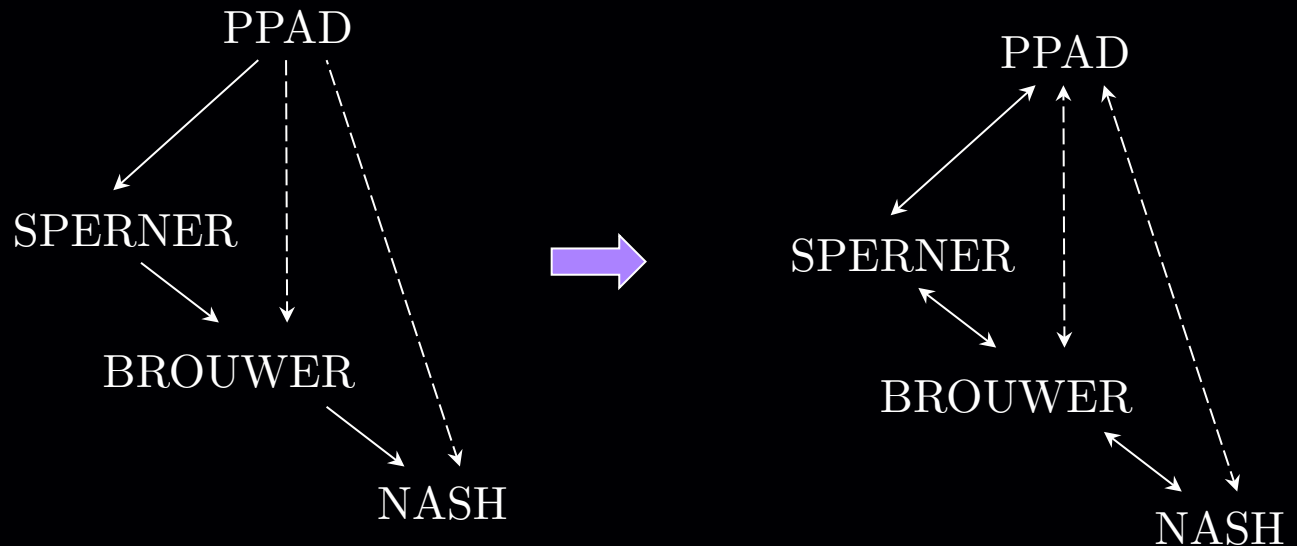
[Mehta'14]: Almost zero-sum games are **PPAD-complete**.

[Chen et al'15]: Anonymous games are **PPAD-complete**.

Reductions following from  
existence proofs:



**[Daskalakis-Goldberg-Papadimitriou'06]:**

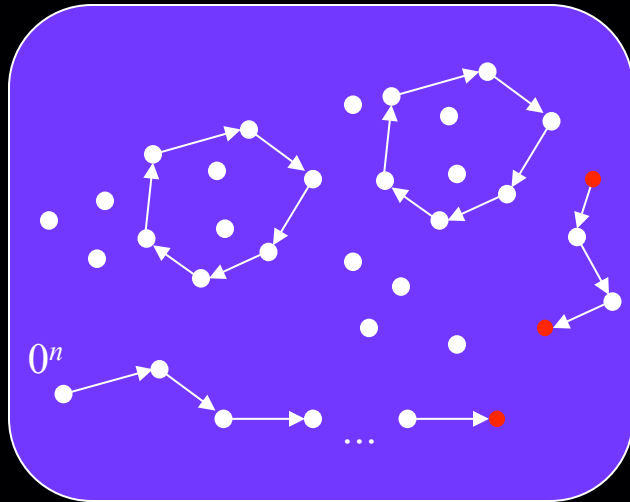


# Menu

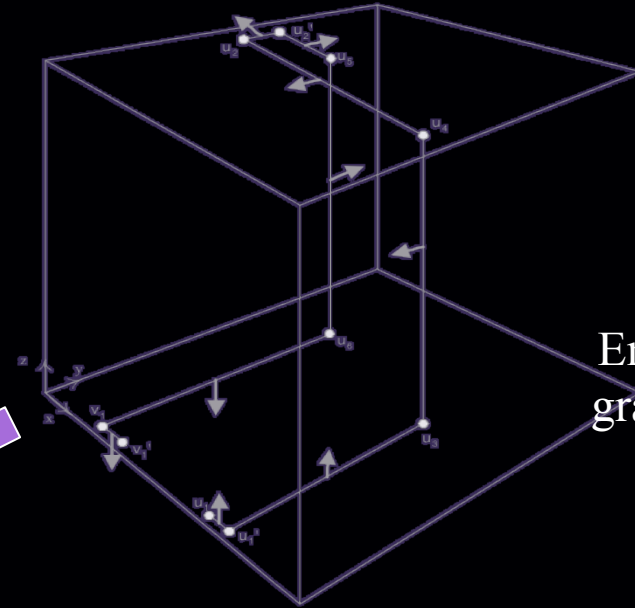
- **Equilibria**
- **Existence proofs**
  - Minimax
  - Nash
  - Brouwer, Brouwer    Nash
  - Sperner, Sperner    Brouwer
- **Complexity of Equilibria**
  - Total Search Problems in NP
  - Proof of Sperner's Lemma
  - PPAD
- **The World Beyond**

# PPAD-Completeness of NASH

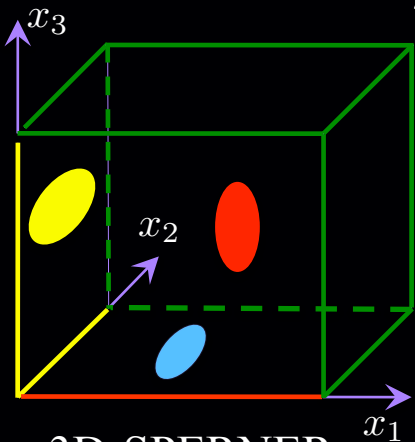
[Daskalakis, Goldberg, Papadimitriou'06]



Generic PPAD



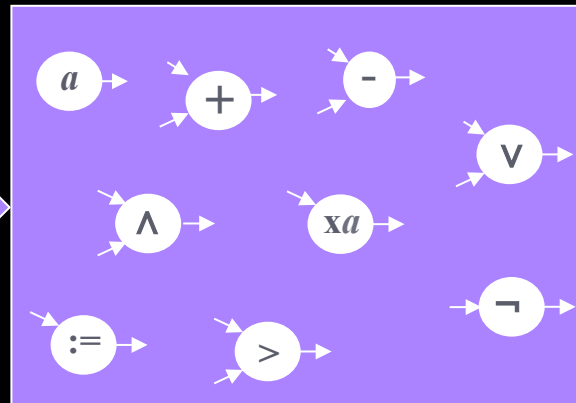
Embed PPAD  
graph in  $[0,1]^3$



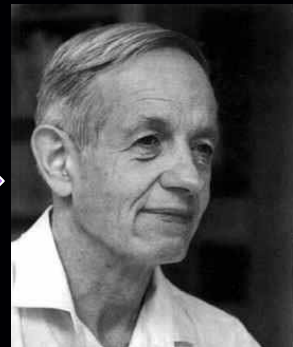
3D-SPERNER



3D-BROUWER



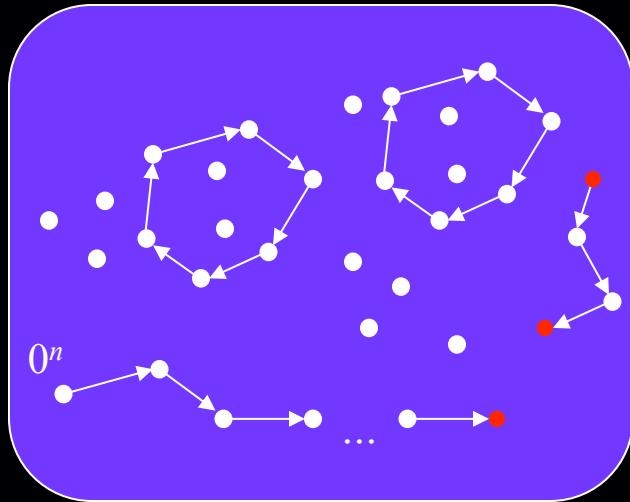
ARITHMCIRCUITSAT



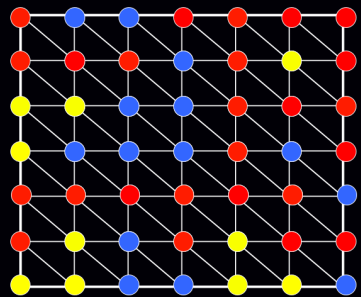
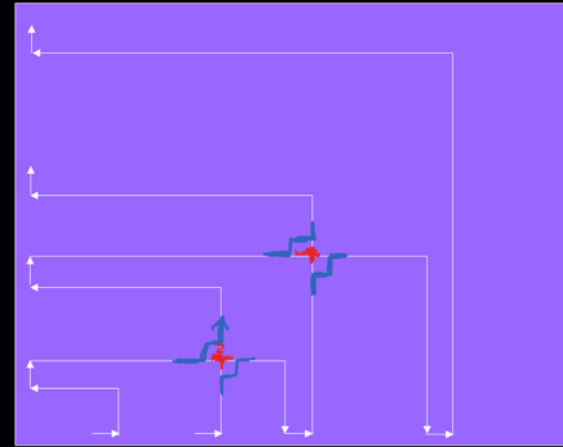
NASH

# PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou'06]



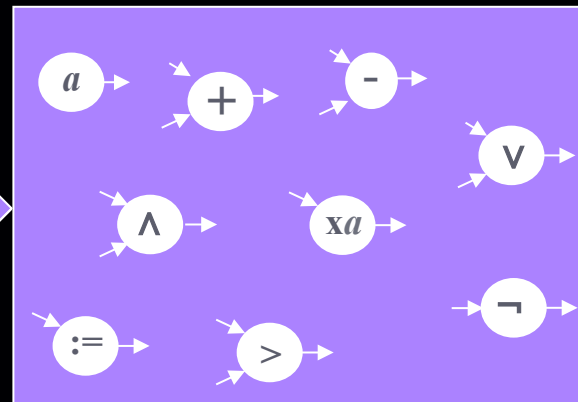
Generic PPAD



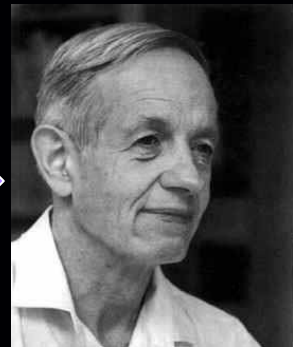
2D-SPERNER



2D-BROUWER



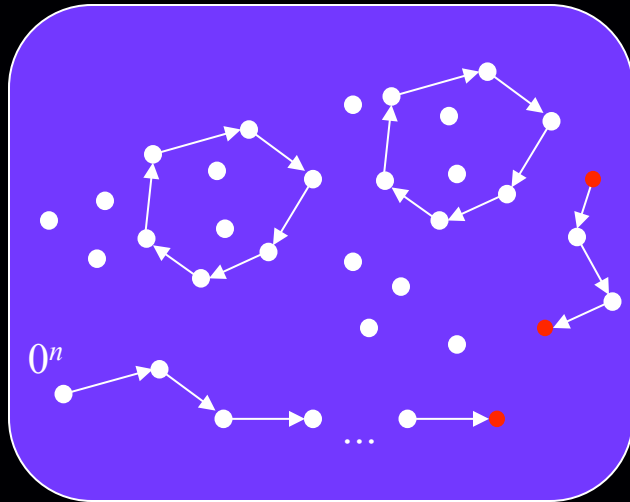
ARITHMCIRCUITSAT



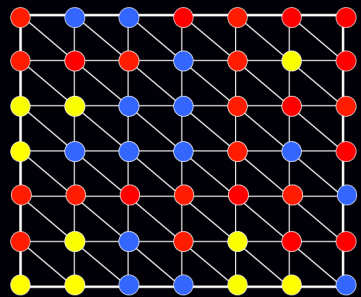
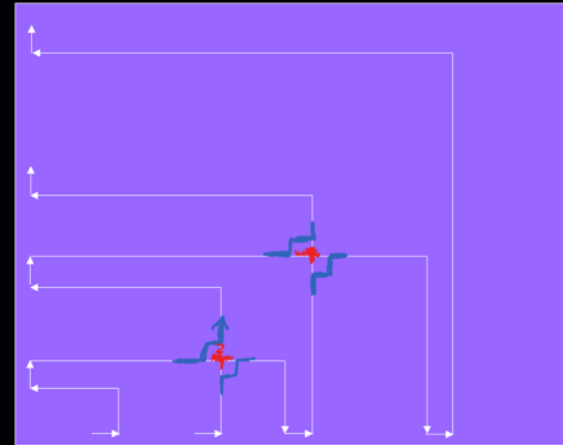
NASH

# PPAD-Completeness of NASH

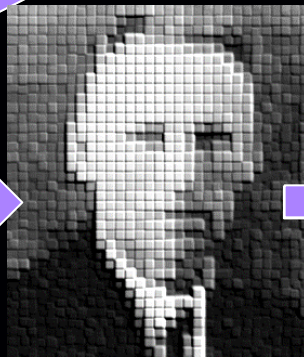
[Daskalakis, Goldberg, Papadimitriou'06]



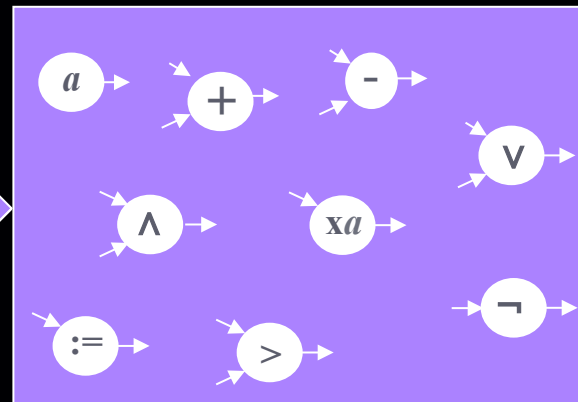
Generic PPAD



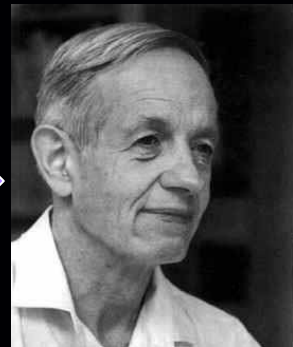
2D-SPERNER



2D-BROUWER



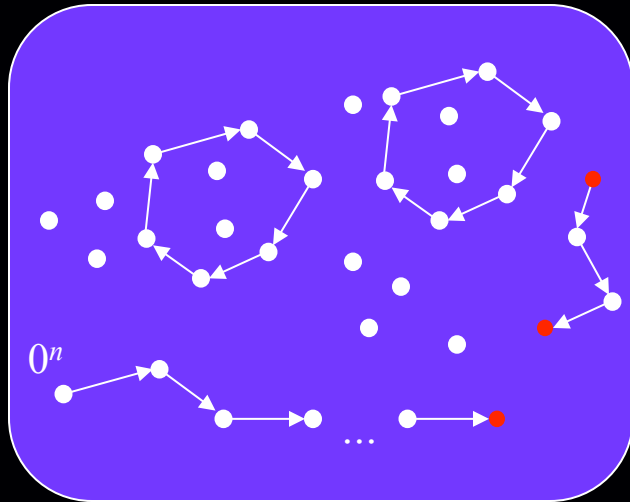
ARITHMCIRCUITSAT



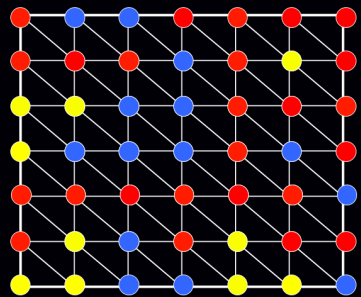
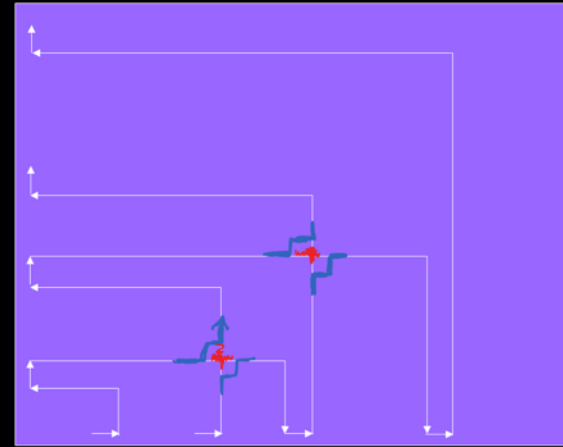
NASH

# PPAD-Completeness of NASH

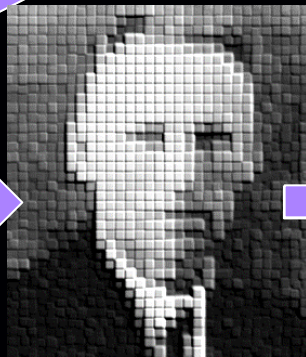
[Daskalakis, Goldberg, Papadimitriou'06]



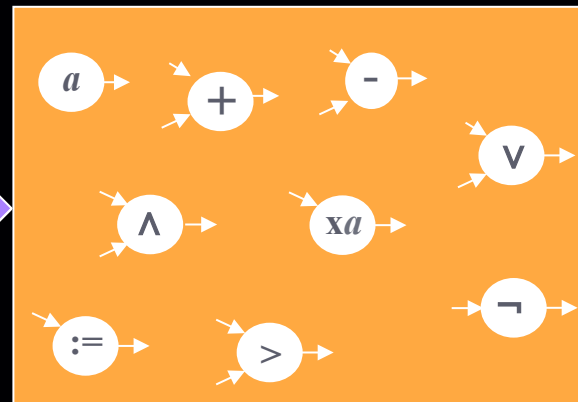
Generic PPAD



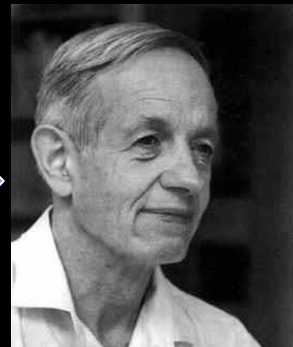
2D-SPERNER



2D-BROUWER



ARITHMCIRCUITSAT









NASH

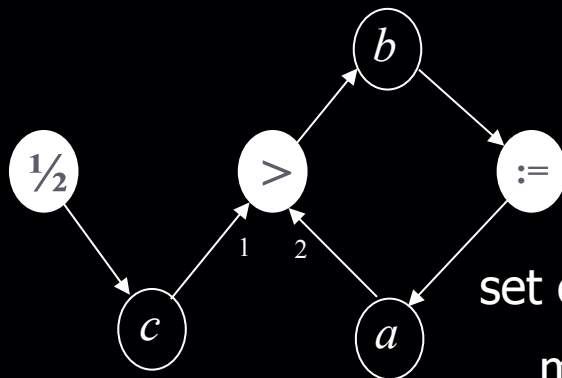
# ARITHMCIRCUITSAT

[Daskalakis, Goldberg, Papadimitriou'06]

INPUT: A circuit comprising:

- variable nodes  $v_1, \dots, v_n$
- gate nodes  $g_1, \dots, g_m$  of types: , , , , , 
- directed edges connecting variables to gates and gates to variables (loops are allowed);
- variable nodes have in-degree 1; gates have 0, 1, or 2 inputs depending on type as above; gates & nodes have arbitrary fan-out

OUTPUT: Values  $v_1, \dots, v_n \in [0,1]$  satisfying the gate constraints:



assignment :  $y == x_1$

addition :  $y == \min\{1, x_1 + x_2\}$

subtraction :  $y == \max\{0, x_1 - x_2\}$

set equal to a constant :  $y == \max\{0, \min\{1, a\}\}$

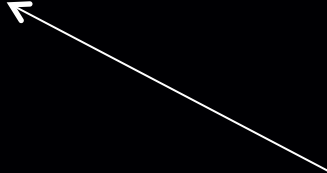
multiply by constant :  $y == \max\{0, \min\{1, a \cdot x_1\}\}$



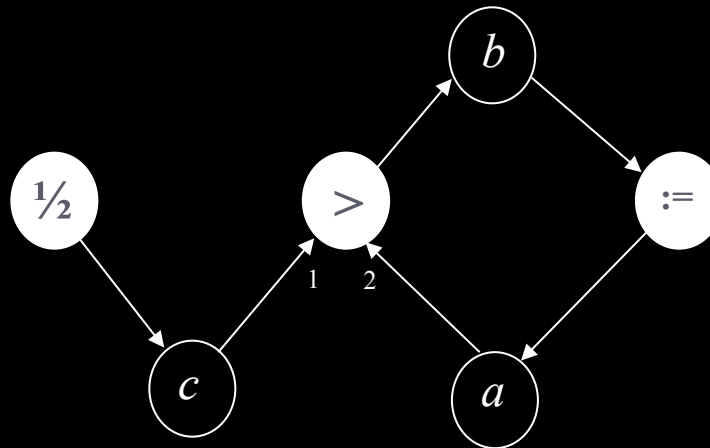
# Comparator Gate Constraints

$$y == \begin{cases} 1, & \text{if } x_1 > x_2 \\ 0, & \text{if } x_1 < x_2 \\ *, & \text{if } x_1 = x_2 \end{cases}$$

any value is allowed



# ARITHMETIC CIRCUIT SAT (example)



Satisfying assignment?

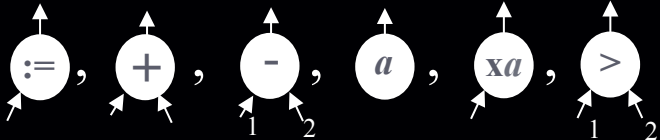
$$a = b = c = \frac{1}{2}$$

# ARITHMCIRCUITSAT

[Daskalakis, Goldberg, Papadimitriou'06]

INPUT: A circuit comprising:


- variable nodes  $v_1, \dots, v_n$

- gate nodes  $g_1, \dots, g_m$  of types: 

- directed edges connecting variables to gates and gates to variables (loops are allowed);

- variable nodes have in-degree 1; gates have 0, 1, or 2 inputs depending on type as above; gates & nodes have arbitrary fan-out

OUTPUT: An assignment of values  $v_1, \dots, v_n \in [0,1]$  satisfying:

  $y == x_1$


[DGP'06]: Always exists satisfying assignment!


  $y == \min\{1, x_1 + x_2\}$

[DGP'06]: but is PPAD-complete to find

  $y == \max\{0, x_1 - x_2\}$

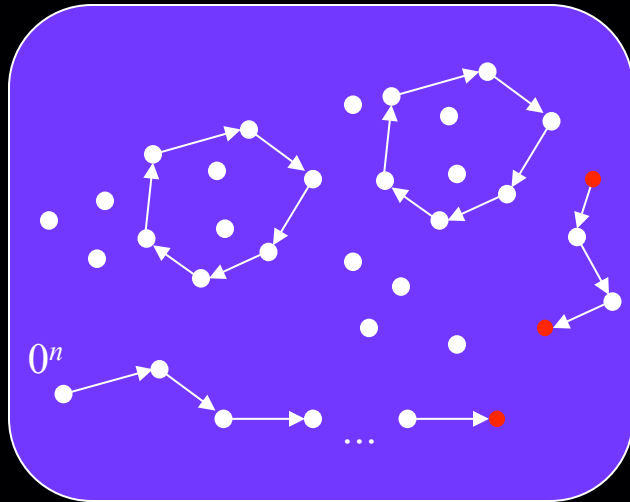
  $y == \max\{0, \min\{1, a\}\}$

  $y == \max\{0, \min\{1, a \cdot x_1\}\}$

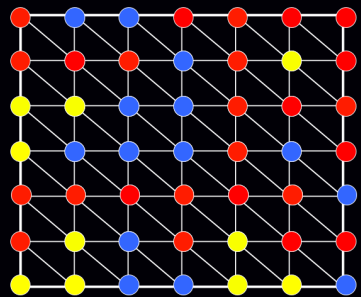
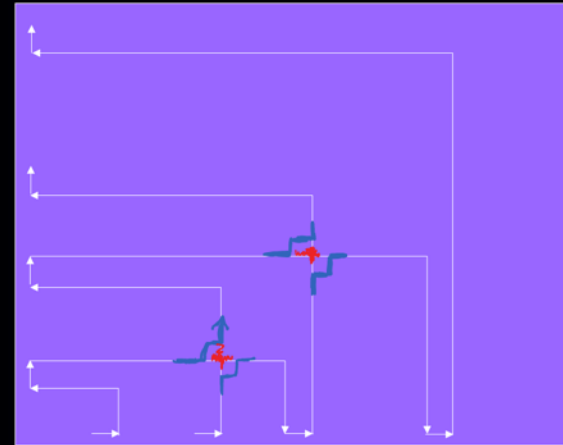
  $y == \begin{cases} 1, & \text{if } x_1 > x_2 \\ 0, & \text{if } x_1 < x_2 \\ *, & \text{if } x_1 = x_2 \end{cases}$

# PPAD-Completeness of NASH

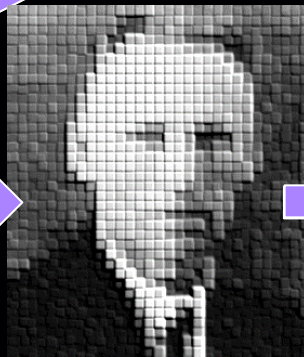
[Daskalakis, Goldberg, Papadimitriou'06]



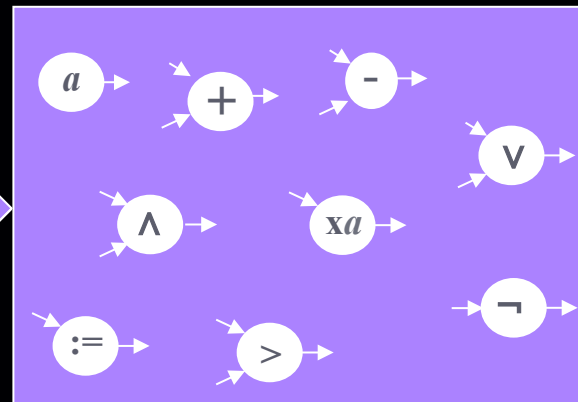
Generic PPAD



2D-SPERNER



2D-BROUWER



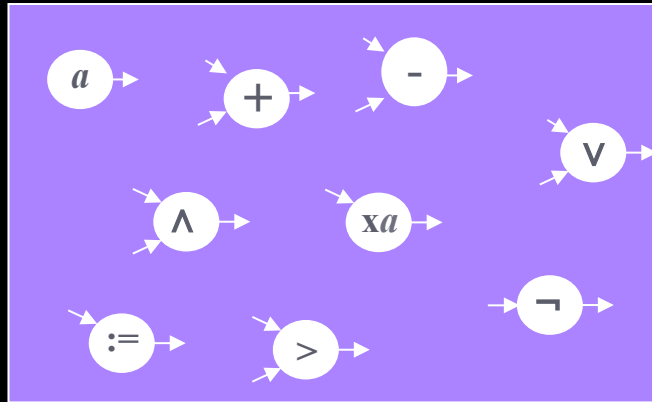
ARITHMCIRCUITSAT



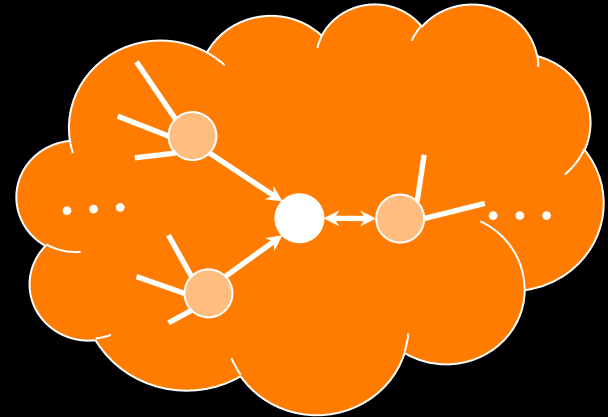
NASH

# PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou'06]



ARITHMCIRCUITSAT



POLYMATRIXNASH

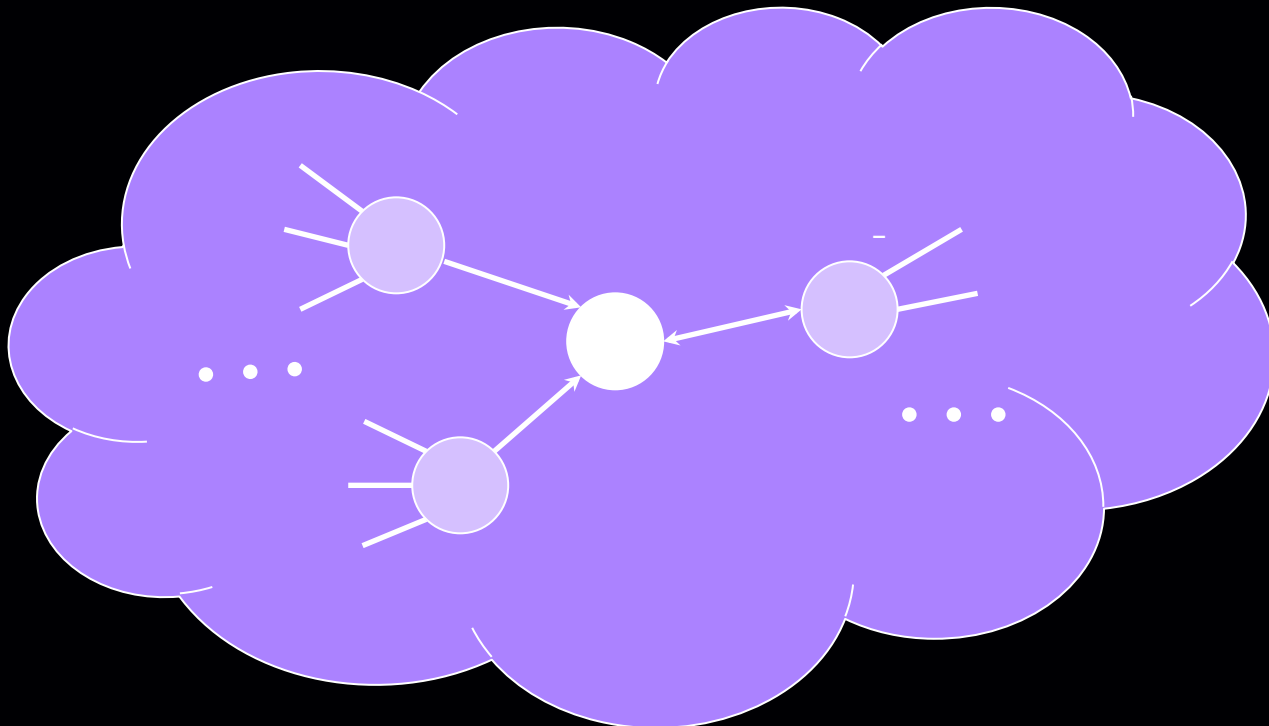


NASH

# Graphical Games

## Graphical Games [Kearns-Littman-Singh'01]

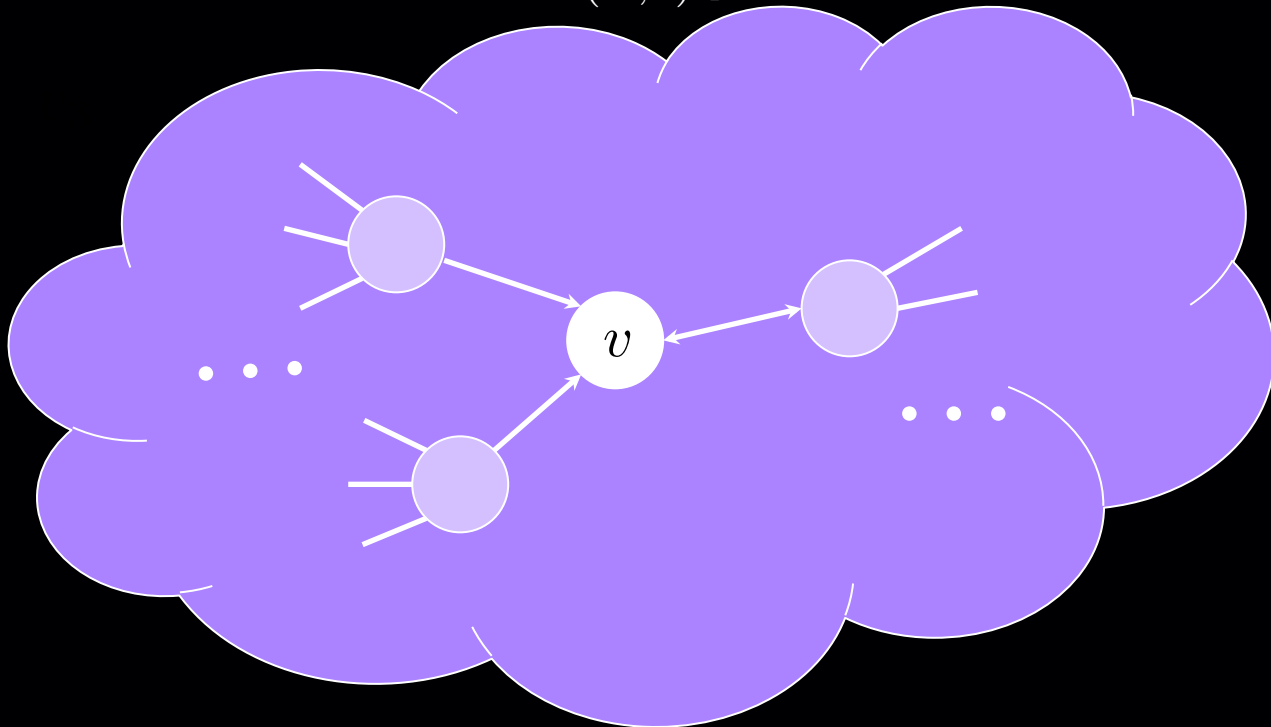
- Defined to capture sparse player interactions, such as those arising under geographical, communication or other constraints.
- Players are nodes in a directed graph.
- Player's payoff only depends on own strategy and the strategies of her in-neighbors in the graph (nodes pointing to her)



# Polymatrix Games

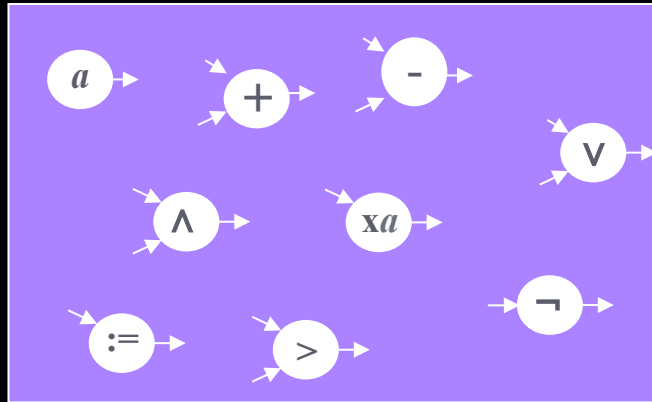
**Polymatrix Games** [Janovskaya'68]: Graphical games with edge-wise separable utility functions.

$$\begin{aligned} u_v(x_1, \dots, x_n) &= \sum_{(w,v) \in E} u_{w,v}(x_w, x_v) \\ &= \sum_{(w,v) \in E} x_v^T A^{(v,w)} x_w \end{aligned}$$

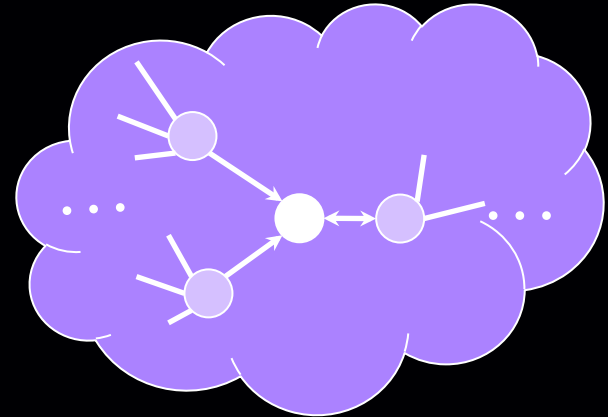


# PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou'06]



ARITHMCIRCUITSAT



POLYMATRIXNASH



NASH



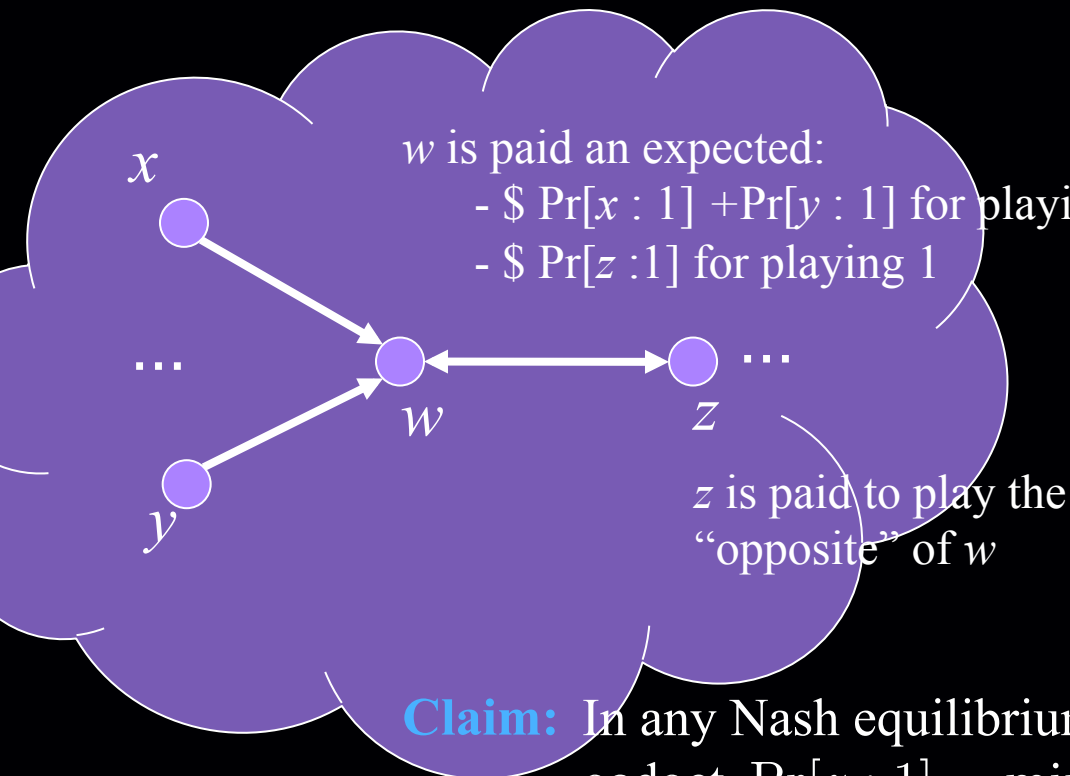
***Game Gadgets:*** Polymatrix games performing real arithmetic at their Nash equilibrium.

# Addition Gadget

Suppose two strategies per player:  $\{0,1\}$

then mixed strategy  $\equiv$  a number in  $[0,1]$  (the probability of playing 1)

e.g. *addition game*



$$\begin{aligned} u(w : 0) &= \Pr[x : 1] + \Pr[y : 1] \\ u(w : 1) &= \Pr[z : 1] \end{aligned}$$

$$\begin{aligned} u(z : 0) &= 0.5 \\ u(z : 1) &= 1 - \Pr[w : 1] \end{aligned}$$

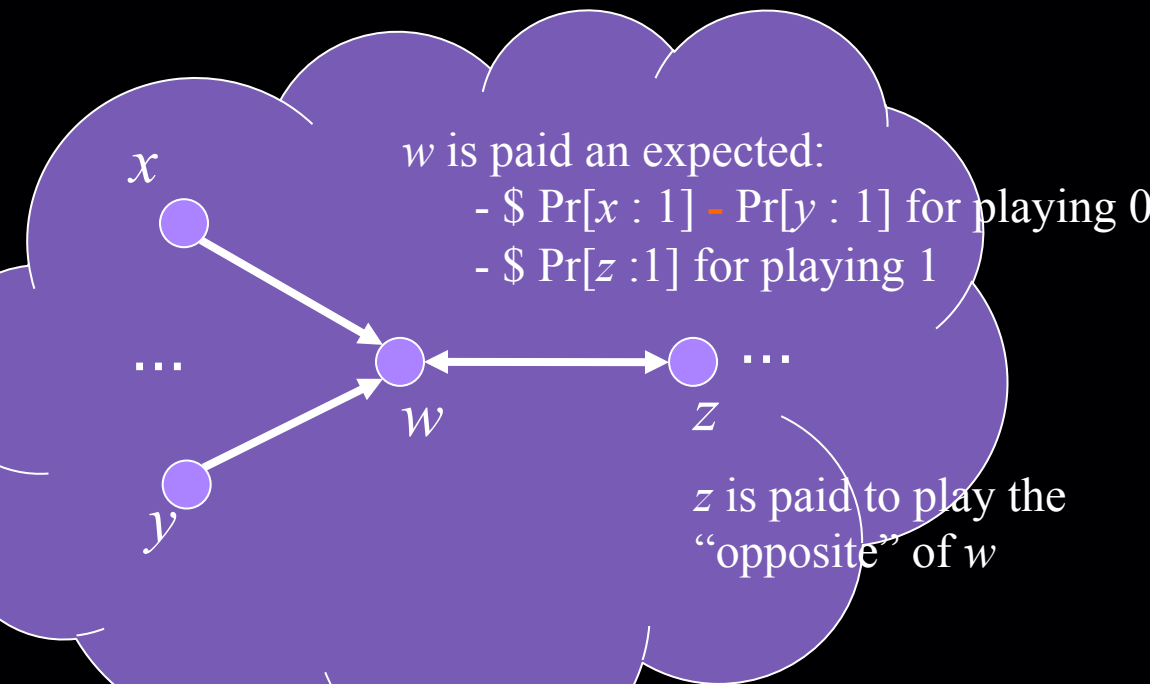
**Claim:** In any Nash equilibrium of a game containing the above gadget  $\Pr[z : 1] = \min\{\Pr[x : 1] + \Pr[y : 1], 1\}$ .

# Subtraction Gadget

Suppose two strategies per player:  $\{0,1\}$

then mixed strategy  $\equiv$  a number in  $[0,1]$  (the probability of playing 1)

e.g. *subtraction*



$$u(w : 0) = \Pr[x : 1] - \Pr[y : 1]$$

$$u(w : 1) = \Pr[z : 1]$$

$$u(z : 0) = 0.5$$

$$u(z : 1) = 1 - \Pr[w : 1]$$

**Claim:** In any Nash equilibrium of a game containing the above gadget  $\Pr[z : 1] = \max\{0, \Pr[x : 1] - \Pr[y : 1]\}$ .

**Notational convention:** Use the name of the player and the probability of that player playing 1 interchangeably.

$$x \quad \curvearrowright \quad \Pr[x : 1]$$

# List of Game Gadgets

copy :  $z = x$

addition :  $z = \min\{1, x + y\}$

subtraction :  $z = \max\{0, x - y\}$

set equal to a constant :  $z = \max\{0, \min\{1, \alpha\}\}$

multiply by constant :  $z = \max\{0, \min\{1, \alpha \cdot x\}\}$

comparison : 
$$z = \begin{cases} 1, & \text{if } x > y \\ 0, & \text{if } x < y \\ *, & \text{if } x = y \end{cases}$$

If any of these gadgets is contained in a bigger polymatrix game, these conditions hold at **any** Nash eq. of that bigger game.

Bigger game can only have edges into the “input players” and out of the “output players.”

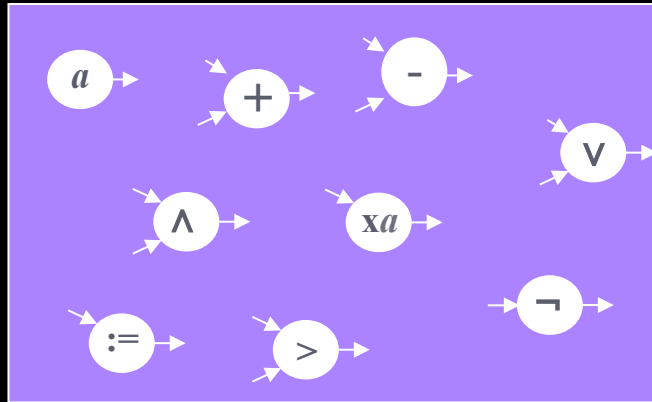
$z$ : “output player” of the gadget

$x, y$ : “input players” of the gadget

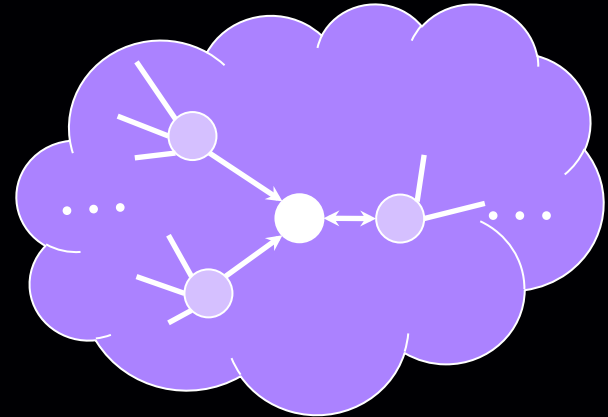
gadgets may have additional players; their graph can be made bipartite

# PPAD-Completeness of POLYMATRIXNASH

[Daskalakis, Goldberg, Papadimitriou'06]



ARITHMCIRCUITSAT



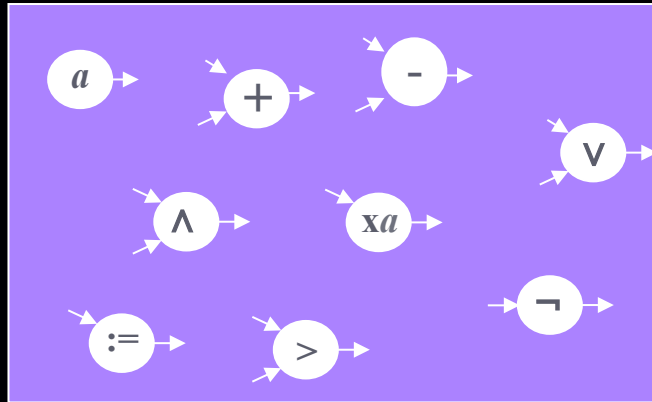
POLYMATRIXNASH

Given arbitrary instance of ARITHMCIRCUITSAT can create polymatrix game by appropriately composing game gadgets corresponding to each of the gates.

At any Nash equilibrium of resulting polymatrix game, the gate conditions are satisfied.

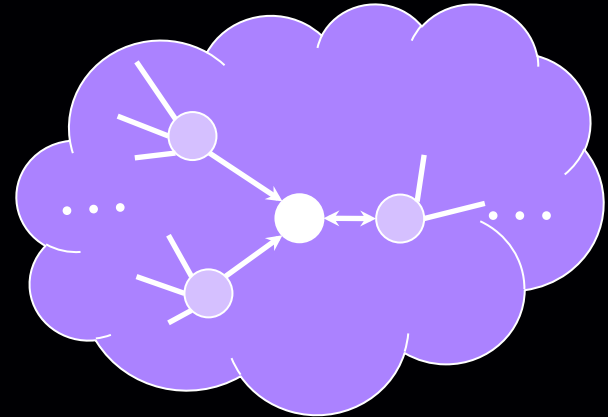
# PPAD-Completeness of NASH

DGP=Daskalakis-Goldberg-Papadimitriou



ARITHMCIRCUITSAT

[DGP'06]



POLYMATRIXNASH

[DGP'06]



[Chen-Deng'06]



4-PLAYER NASH



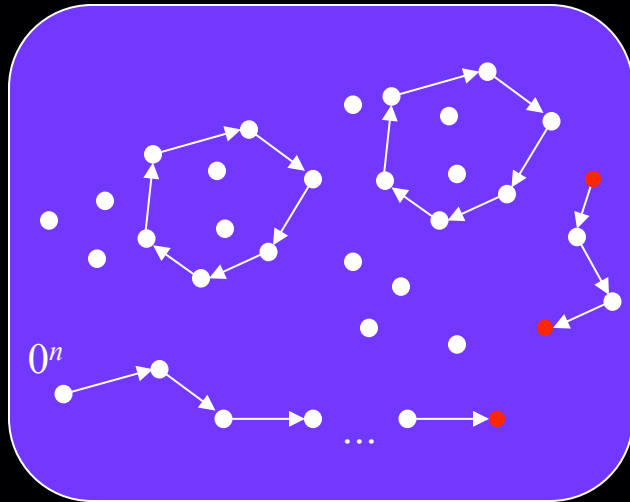
3-PLAYER NASH



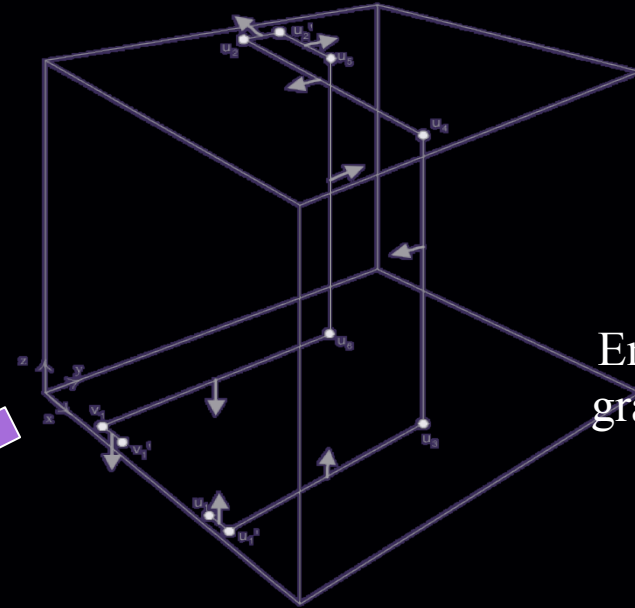
2-PLAYER NASH

# PPAD-Completeness of NASH

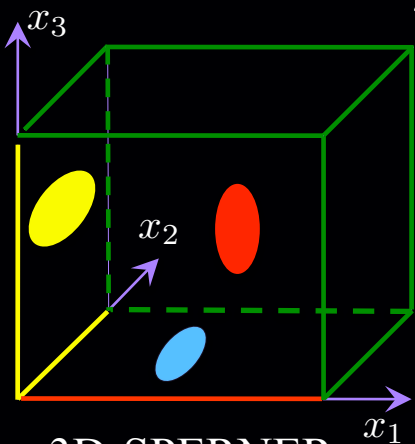
[Daskalakis, Goldberg, Papadimitriou'06]



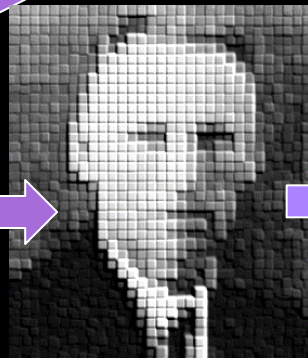
Generic PPAD



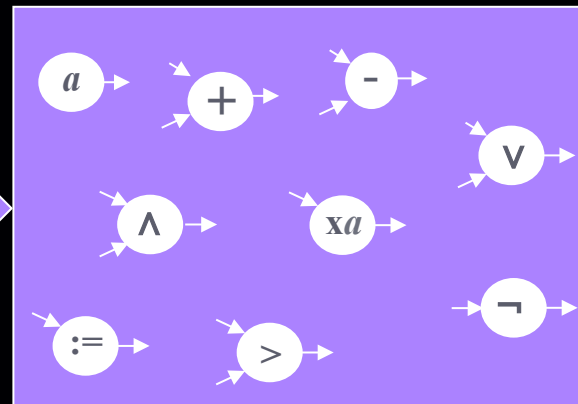
Embed PPAD graph in  $[0,1]^3$



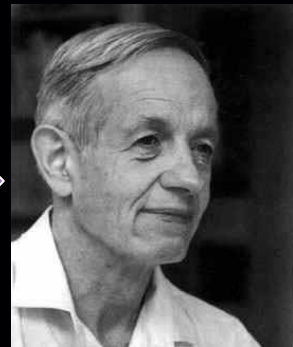
3D-SPERNER



3D-BROUWER



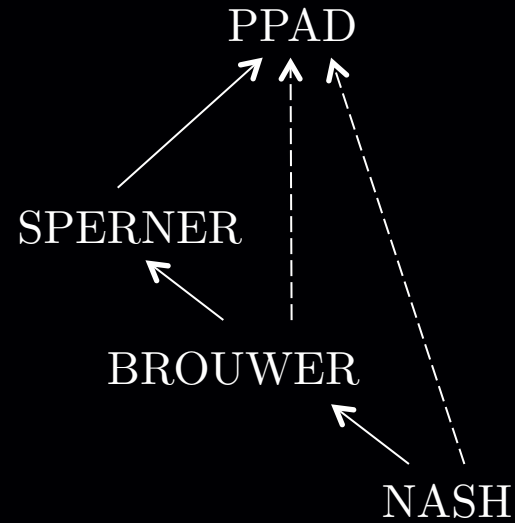
ARITHMCIRCUITSAT



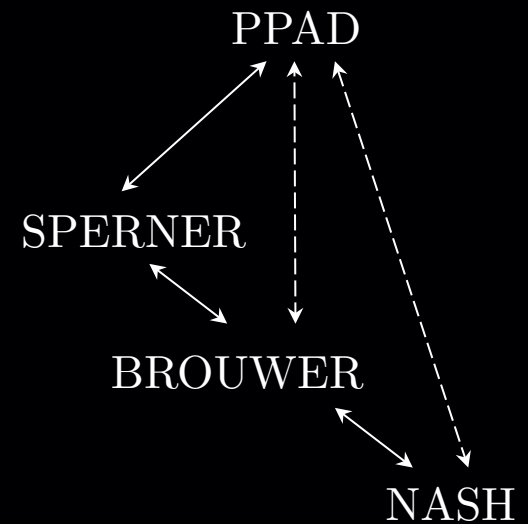
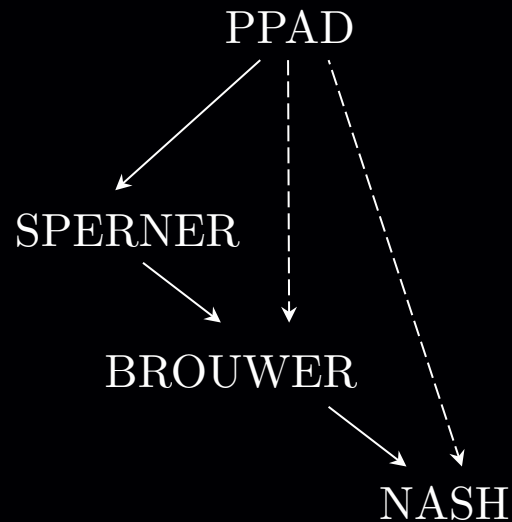
NASH



Classical Inclusions:



[Daskalakis-Goldberg-Papadimitriou'06]:



# The Complexity of Nash Equilibrium

[Daskalakis, Goldberg, Papadimitriou '06]: Finding a Nash equilibrium is **PPAD-complete**.

[Chen, Deng'06]: ...even in two-player games.

I.e. finding a Nash equilibrium is *computationally intractable*, exactly as intractable as the class **PPAD**, SPERNER, BROUWER

[Condenotti et al'06,...,Chen et al'13]: Arrow-Debreu equilibria (in markets w/ complements) are also **PPAD-hard**.

[Mehta'14]: Almost zero-sum games are **PPAD-complete**.

[Chen et al'15]: Anonymous games are **PPAD-complete**.

**Markets**



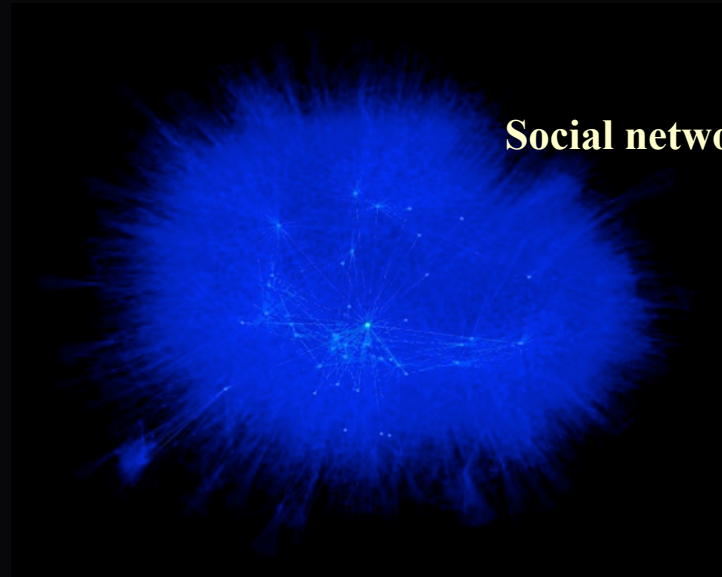
**Traffic**



**Evolution**



**Social networks**



Robert Aumann, 1987:

*“Two-player zero-sum games are one of the few areas in game theory, and indeed in the social sciences, where a fairly sharp, unique prediction is made.”*

— Indeed equilibria of zero-sum games are efficiently computable, comprise a convex set, can be reached via dynamics efficiently

— While outside of zero-sum games equilibria are PPAD-complete, disconnected, and not reachable via dynamics

GAME OVER ?

absolutely NOT!

# Escape 1: Approximation

- ▶ Maybe Nash equilibrium is hard to compute, but *approximate equilibria* are tractable.
  - no player has no more than some small  $\epsilon$  incentive to deviate.
- ▶ Absolute vs Relative Approximation?
- ▶ Relative (standard in CS):
- ▶ [Daskalakis'11, Rubinstein'15]: For some  $\epsilon > 0$ , in 2-player games, computing a pair of mixed strategies s.t. no player can improve his current payoff by more than an  $\epsilon$ -fraction is **PPAD**-complete.
- ▶ Absolute Error (standard for fixed points-[Scarf'67])?
  - We know that the problem is unlikely **PPAD**-hard [Lipton-Markakis-Mehta'03, Barman'15]: finding  $\epsilon$ -Nash of 2-player  $n$ -strategy game in  $n \log n / \epsilon^2$  time
  - Polynomial time algorithm is missing despite a long line of research [Kontogiannis et al '06], [Daskalakis et al'06, '07], [Bosse et al'07], [Tsaknakis, Spirakis'08]....?

# Escape 2: Games w/ Special Structure

- ▶ Arbitrary normal form are hard, but 2-player zero-sum aren't.
- ▶ Identify even broader families of games that are tractable.
- ▶ Upshot:
  - Whenever game of interest has special structure, can be more confident on equilibrium predictions & can actually find them.
  - When designing a new game, design it so that it has this good special structure.
- ▶ Two examples:
  - Multi-player Zero-Sum Games
  - Anonymous Games



# Escape 2: Games w/ Special Structure

- ▶ Arbitrary normal form are hard, but 2-player zero-sum aren't.
- ▶ Identify even broader families of games that are tractable.
- ▶ Upshot:
  - Whenever game of interest has special structure, can be more confident on equilibrium predictions & can actually find them.
  - When designing a new game, design it so that it has this good special structure.
- ▶ Two examples:
  - **Multi-player Zero-Sum Games**
  - Anonymous Games



# Multiplayer Zero-Sum...what?

Take an arbitrary two-player game, between Alice and Bob.

Add a third player, Sam, who does not affect Alice or Bob's payoffs, but receives payoff:

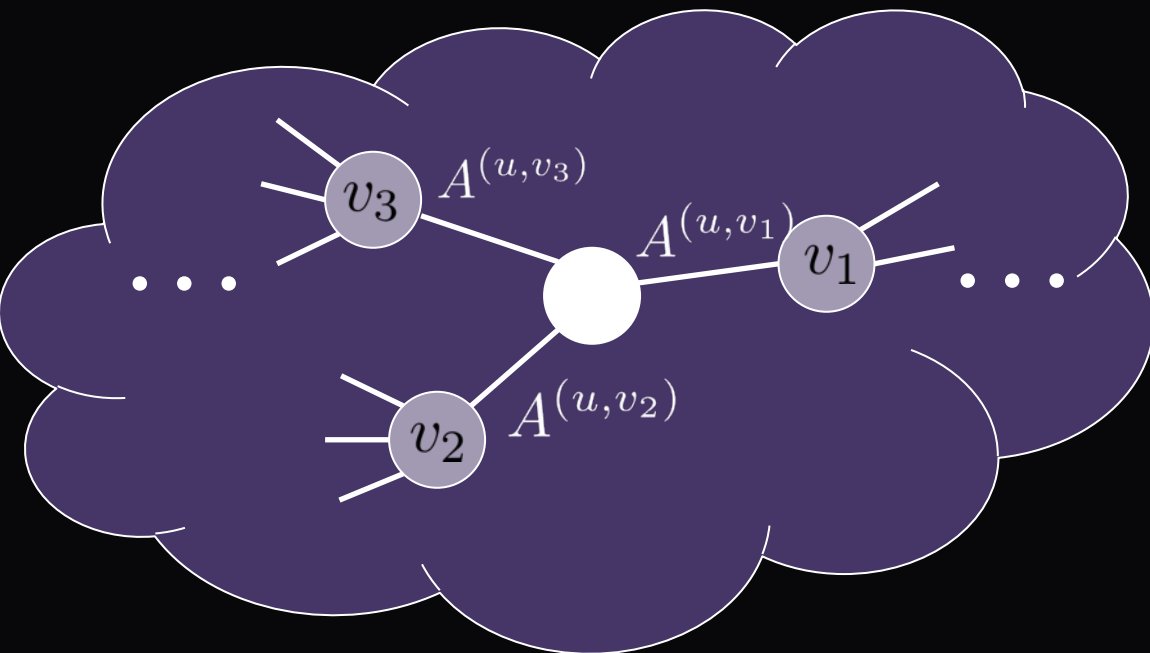
$$-(P \downarrow Alice(\sigma) + P \downarrow Bob(\sigma)), \forall \text{ outcomes } \sigma$$

The game is zero-sum, but solving it is **PPAD**-complete.

➡ intractability even for 3 players, if *three-way* interactions allowed.

What if only *pairwise* interactions are allowed?

# Zero-Sum Polymatrix Games



- players are nodes of a graph  $G$
- edges are 2-player games
- each player's payoff is the sum of payoffs from all adjacent edges:

$$\sum_{i=1}^n x_i u_i^T A(u, v_i) x_i v_i$$

N.B. Finding a Nash equilibrium is **PPAD**-complete [Daskalakis, Goldberg, Papadimitriou '06]; even constant approximations are [Rubinstein'15]

*But what if the game is zero-sum, i.e. the sum of all players' payoffs is 0?*

# Zero-Sum Polymatrix Games (cont.)

[Daskalakis, Papadimitriou '09; Cai, Daskalakis'10; Cai, Candogan, Daskalakis, Papadimitriou'15]:

In zero-sum polymatrix games:

- a Nash equilibrium can be found efficiently with linear-programming
- the Nash equilibria comprise a convex set
- if every node uses a no-regret learning algorithm, the players' behavior converges to a Nash equilibrium
  - empirical strategies approach Nash equilibrium
- *I.e. several good properties of two-player zero-sum games are inherited.*

# Escape 2: Games w/ Special Structure

- ▶ Arbitrary normal form are hard, but 2-player zero-sum aren't.
- ▶ Identify even broader families of games that are tractable.
- ▶ Upshot:
  - Whenever game of interest has special structure, can be more confident on equilibrium predictions & can actually find them.
  - When designing a new game, design it so that it has this good special structure.
- ▶ Two examples:
  - Multi-player Zero-Sum Games
  - **Anonymous Games**

# Anonymous Games

- ▶ **Anonymous Game:** Every player might have a different payoff function, which only depends symmetrically on the other players' actions.
  - e.g. auction, traffic, social phenomena---see e.g. “The women of Cairo: Equilibria in Large Anonymous Games.” by Blonski, GEB’99.
- ▶ **[Daskalakis-Papadimitriou’07-’09, Daskalakis-Kamath-Tzamos’15]:** Arbitrarily good approximations are tractable if #strategies does not scale to infinity.
- ▶ Recall **[Chen et al’15]:** Exact equilibria are intractable.
- ▶ Interesting relation to limit theorems in probability.
- ▶ E.g. “ $\forall \epsilon, n$ , the sum  $X_1 + \dots + X_n$  of arbitrary independent Bernoulli 0/1 random variables is  $\epsilon$ -close in  $\ell_1$  distance to
  - the sum of i.i.d. Bernoullis; or
  - $C + \sum_{i=1}^n 1/\epsilon^3 \cdot Y_i$ , for some constant  $C$  and independent Bernoullis  $Y_1, \dots, Y_n$  ”
- ▶ Implies: “In every  $n$ -player 2-strategy anonymous game, there exists  $\epsilon$ -Nash

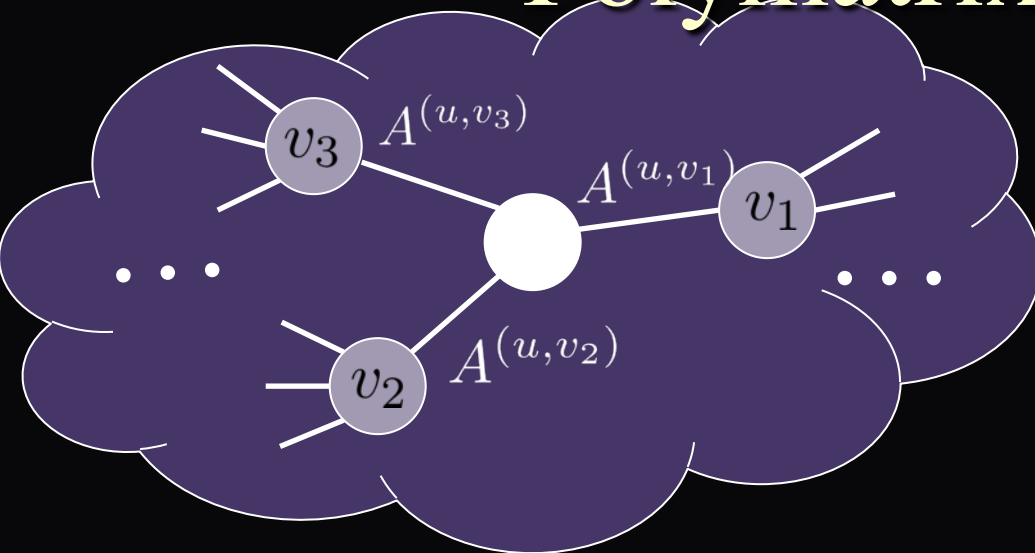
# Escape 3: Alternative Solution Concepts

- ▶ If Nash equilibrium is intractable for a family of games, chances are it is not always discovered by players playing those games.
- ▶ So focus on alternatives that are tractable and thereby more plausible.
- ▶ Two canonical and plausible alternatives:
  - *Correlated equilibrium*:
    - ▶ generalizes Nash equilibrium, and is tractable
  - *No-regret learning behavior*:
    - ▶ Natural way to axiomatize player dynamical behavior
    - ▶ Strong connection to learning, online optimization
    - ▶ Generalizes correlated equilibrium (limits: coarse corr. Eq)

# Correlated vs Nash

- ▶ Won't give formal definition of correlated equilibrium.
  - similar to Nash, except players' randomization can be correlated
  - “no player has incentive to deviate given own sampled pure action from the joint distribution”
- ▶ Equilibrium conditions expressible as linear constraints on the joint action distribution.
  - hence solvable via linear program
- ▶ In normal form games, linear program has polynomial size in the game description:
  - the LP maintains a variable for every pure strategy profile
  - Same #variables as total #payoff entries required to specify game
- ▶ So correlated eq in **P**, while Nash is **PPAD**-complete.

# Polymatrix Games



- Players are nodes of a graph  $G$
- Edges are 2-player games
- Each player's payoff is the sum of payoffs from all adjacent edges; e.g.  

$$\sum_{i=1}^n \sum_{j=1}^n x_i u_i^T A(u, v_j) x_j v_j^T$$

- Description size:  $2 \cdot m \cdot s^2$  (two payoff tables/edge)  $\leq n^2 \cdot s^2$  payoff entries
- Joint dist'n over players' actions:  $s^n$  probabilities (one per pure strategy profile)
- So size of correlated equilibrium exponentially larger than size of game  
 $\Rightarrow$  computing it seems hopeless...

**[Papadimitriou, Roughgarden'05; Jiang, Leyton-Brown'10]:** Poly-time algorithm.

Crucial Idea: #correlated eq constraints  $\leq n \cdot s$  (one per player-action pair)

Use dual LP, properties of Ellipsoid algorithm.

Extends to any game where expected payoffs under independent mixed strategies can be computed in polynomial-time.



# Summary: Equilibrium Complexity

- ▶ Equilibria may be computationally intractable.
  - Nash equilibria in normal form games are **PPAD**-complete
  - Same is true of many equilibria in economics
- ▶ When intractable, their universality is questionable.
  - Cannot hope that players always discover them.
  - Analyst cannot count on always finding them.
- ▶ Important to identify game families where equilibria are tractable.
  - several classes of tractable games have been found, e.g. polymatrix zero-sum, anonymous
- ▶ Consider alternative solution concepts with better computational properties, e.g. correlated eq, no-regret learning.
- ▶ Understand the complexity of approximate solution concepts.
- ▶ Investigating the complexity of equilibria offered complexity theory challenging problems that enriched the field and continue providing interesting challenges going forward.

# Menu

- **Equilibria**
- **Existence proofs**
  - Minimax
  - Nash
  - Brouwer, Brouwer    Nash
  - Sperner, Sperner    Brouwer
- **Complexity of Equilibria**
  - Total Search Problems in NP
  - Proof of Sperner's Lemma
  - PPAD
- **The World Beyond**

# Menu

- **Equilibria**
- **Existence proofs**
  - Minimax
  - Nash
  - Brouwer, Brouwer    Nash
  - Sperner, Sperner    Brouwer
- **Complexity of Equilibria**
  - Total Search Problems in NP
  - Proof of Sperner's Lemma
  - PPAD
- **The Complexity World Beyond: between P and NP**

# Other arguments of existence, and resulting complexity classes

“If a graph has a node of odd degree, then it must have another.”

**PPA**

“Every directed acyclic graph must have a sink.”

**PLS**

“If a function maps  $n$  elements to  $n-1$  elements, then there is a collision.”

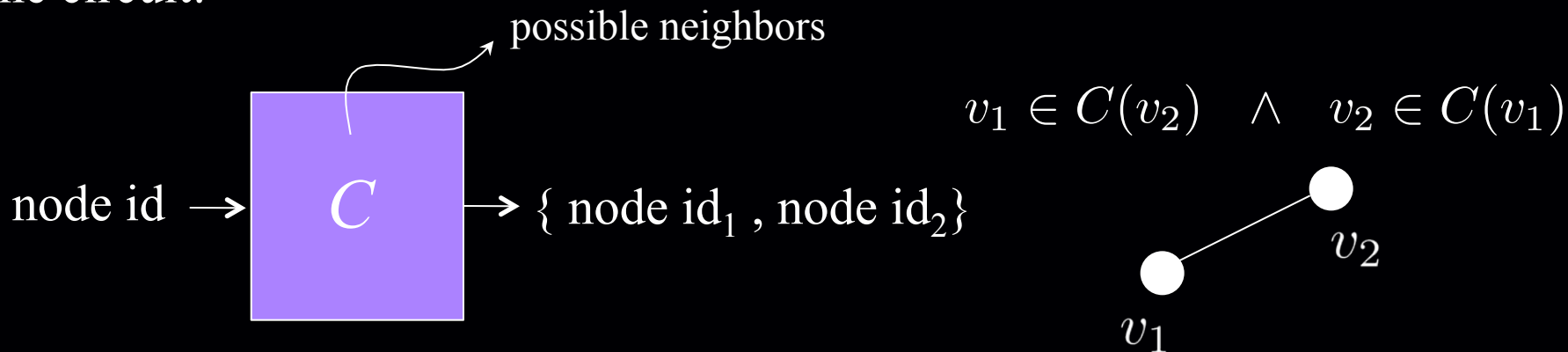
**PPP**

Formally?

# The Class PPA [Papadimitriou '94]

*“If a graph has a node of odd degree, then it must have another.”*

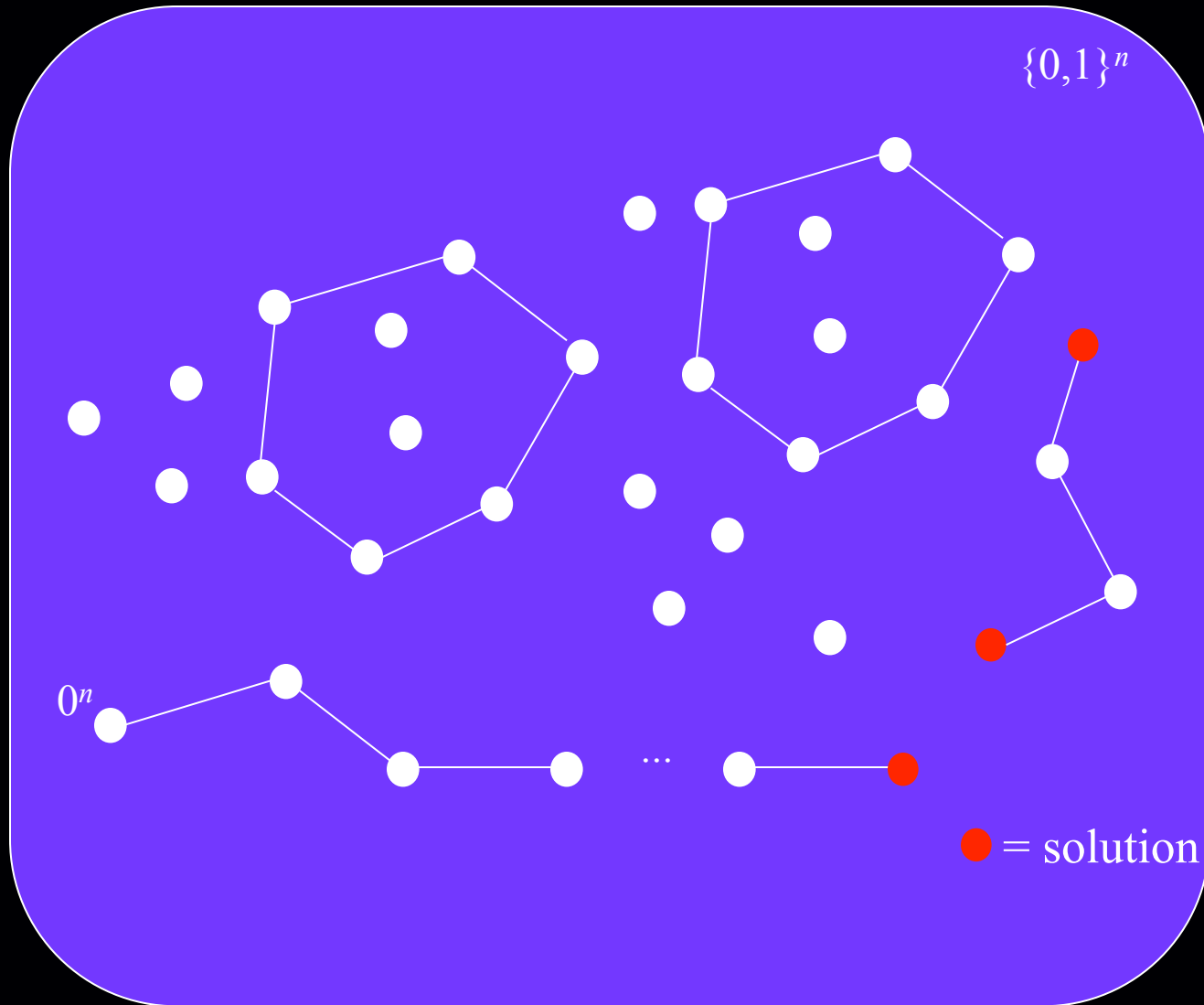
Suppose that an exponentially large graph with vertex set  $\{0,1\}^n$  is defined by one circuit:



**ODDDEGREE NODE:** Given  $C$ : If  $0^n$  has odd degree, find another node with odd degree. Otherwise say “yes.”

**PPA =** { Search problems in FNP reducible to ODDDEGREE NODE }

# ODDDEGREE NODE



# SMITH $\in$ PPA

**SMITH:** Given Hamiltonian cycle in 3-regular graph and an edge that it uses, find another one.

**[Smith]:** There must be another one.

