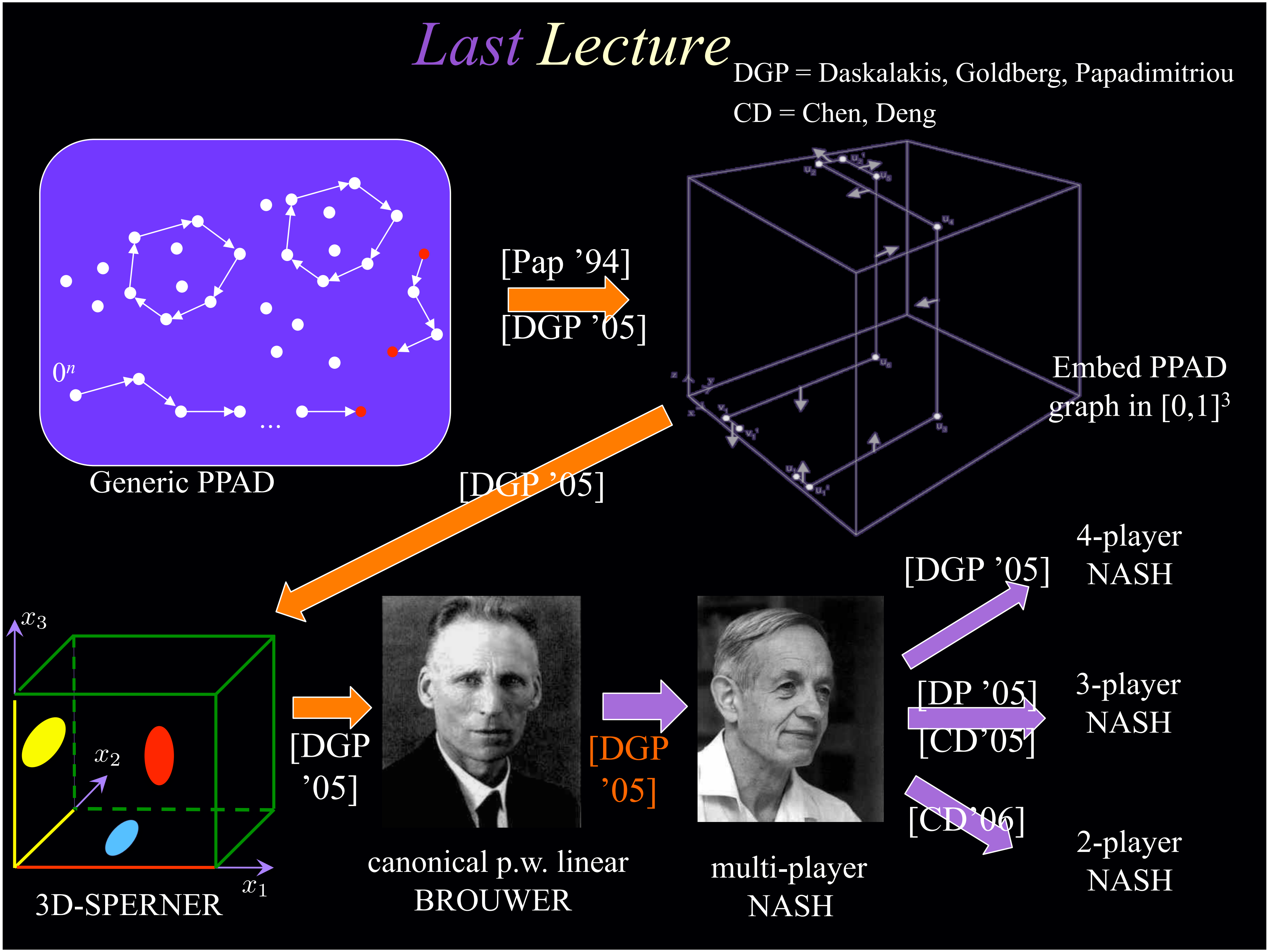
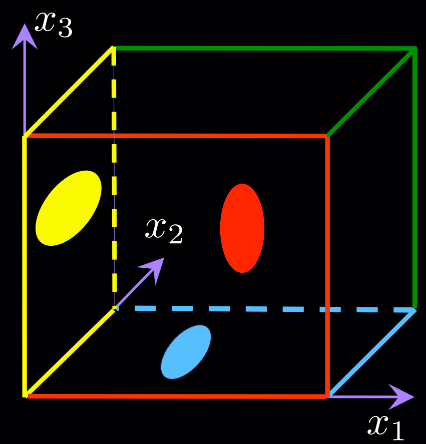
6.896: Topics in Algorithmic Game Theory

Lecture 10

Constantinos Daskalakis


Last Lecture
DGP = Daskalakis, Goldberg, Papadimitriou
CD = Chen, Deng
0^n
...
Generic PPAD
[Pap '94]
[DGP '05]
Embed PPAD graph in $[0,1]^3$
[DGP '05]
x_3
x_2
x_1
3D-SPERNER
[DGP '05]
canonical p.w. linear BROUWER
[DGP '05]
multi-player NASH
[DGP '05]
4-player NASH
[DP '05]
[CD'05]
3-player NASH
[CD'06]
2-player NASH

Canonical BROUWER instance



- Partition every dimension into multiples of 2^{-m}.

- Using the SPERNER coloring (which itself was obtained via the embedding of the PPAD graph into $[0,1]^3$), define at the center of each cubelet one of 4 possible displacement vectors

color 0 (ambient space) $(-1,-1,-1) \times \alpha$

color 1 $(1,0,0) \times \alpha$

color 2 $(0,1,0) \times \alpha$

color 3 $(0,0,1) \times \alpha$

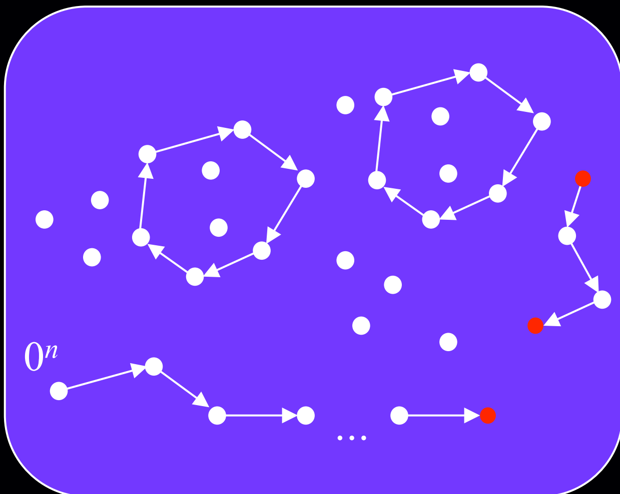
$$\boxed{\alpha = 2^{-2m}}$$

- The goal is to find a point of the subdivision s.t. among the 8 cubelets containing it, all 4 displacements are present.

This Lecture

DGP = Daskalakis, Goldberg, Papadimitriou

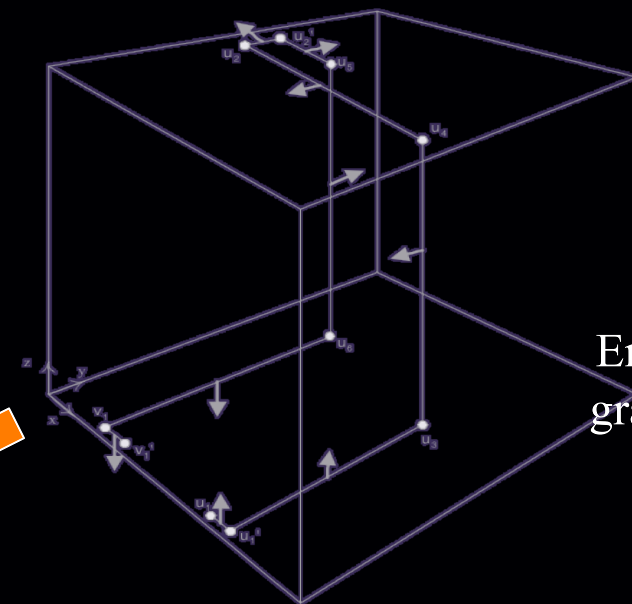
CD = Chen, Deng



Generic PPAD

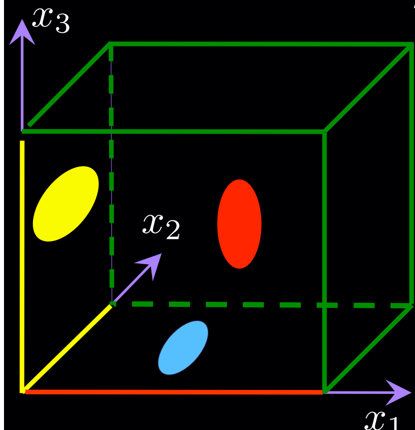
[Pap '94]

[DGP '05]



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

[DGP '05]

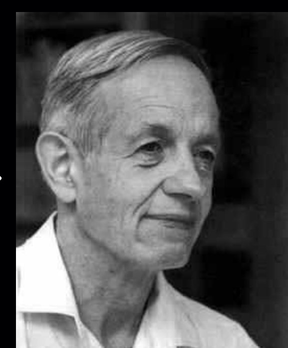


3D-SPERNER

[DGP '05]

p.w. linear BROUWER

[DGP '05]



multi-player NASH

[DGP '05]

4-player NASH

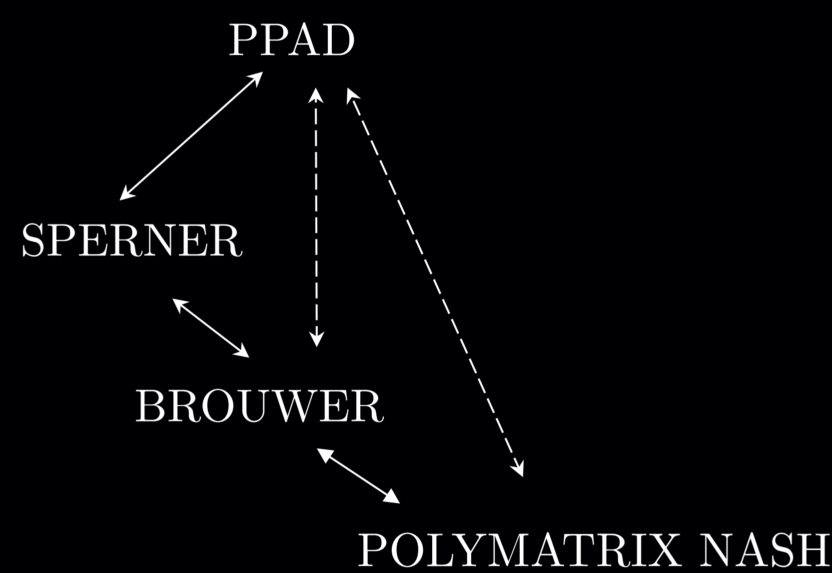
[DP '05]

[CD'05]

3-player NASH

[CD'06]

2-player NASH

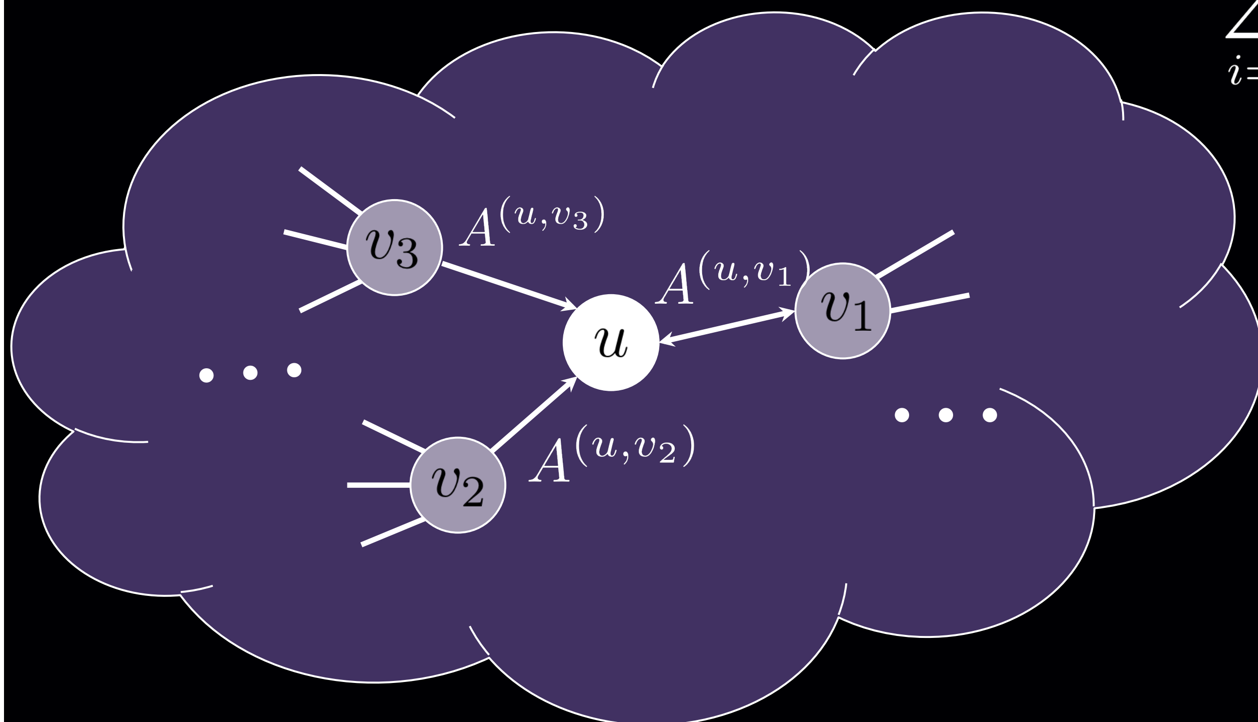

PPAD
SPERNER
BROUWER
POLYMATRIX NASH

Polymatrix Games

Graphical games with edge-wise separable utility functions.

- edges are 2-player games
- player's payoff is the sum of payoffs from all adjacent edges

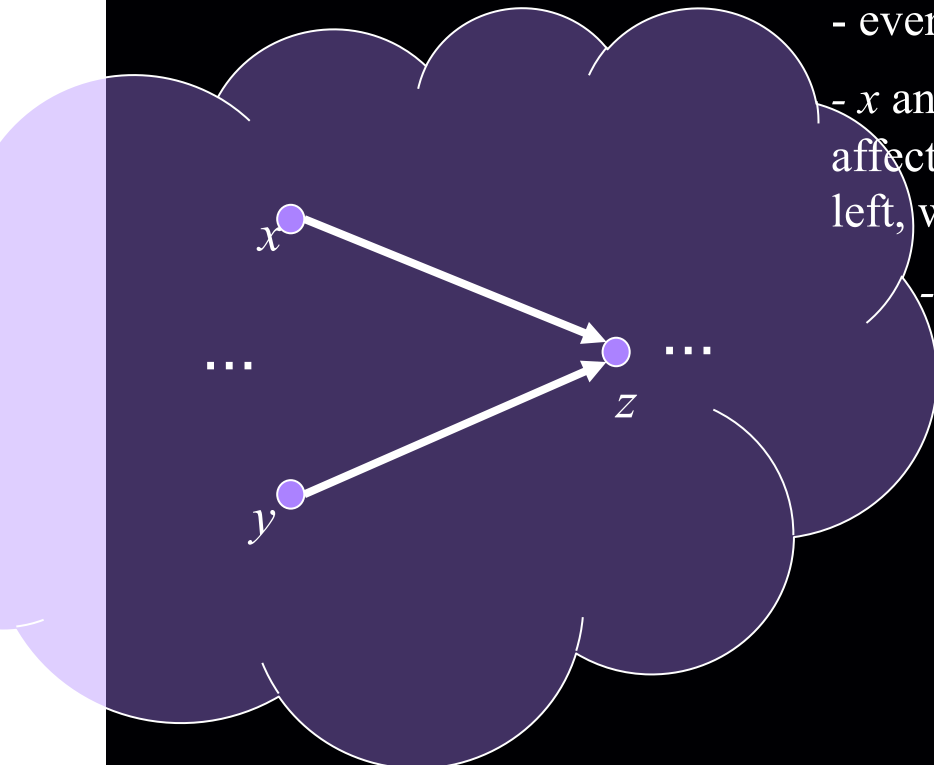
$$\sum_{i=1}^{3} x_u^{\mathrm{T}} A^{(u,v_i)} x_{v_i}$$



Game Gadgets

Binary computations

- 3 players: x, y, z
  (imagine they are part of a larger graphical game)



- every player has strategy set {0, 1}
- x and y do not care about z , i.e. their strategies are affected by the larger game containing the game on the left, while z cares about x and y
- z's payoff table:

separable

$z : 0$

	$y : 0$	$y : 1$
$x : 0$	1	0.5
$x : 1$	0.5	0

$z : 1$

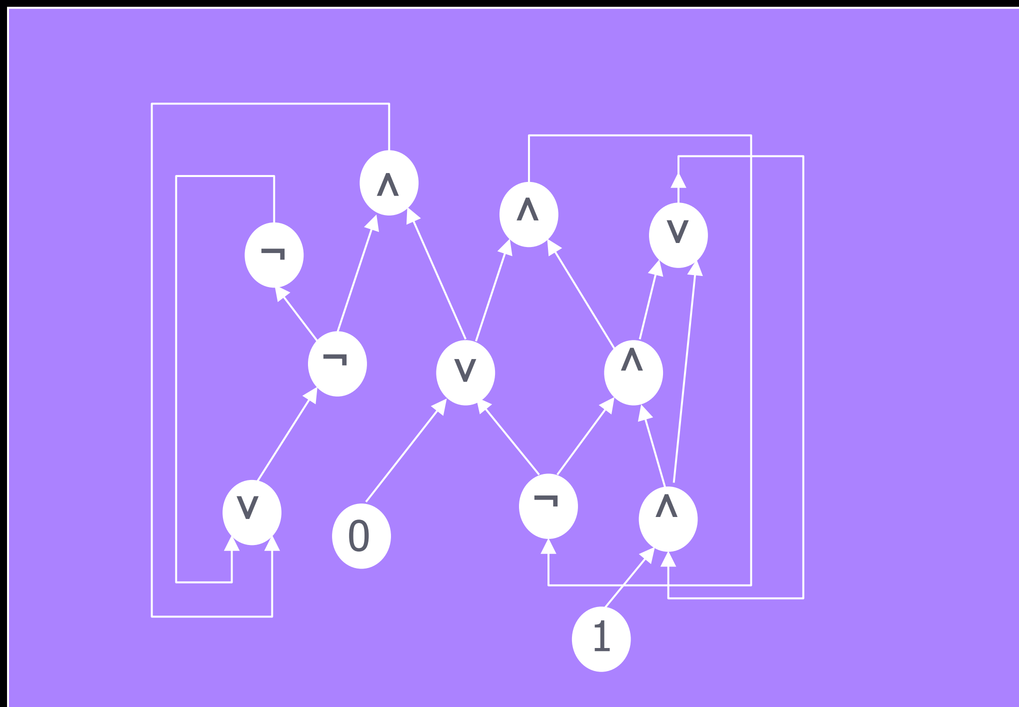
	$y : 0$	$y : 1$
$x : 0$	0	1
$x : 1$	1	2

Claim: In any Nash equilibrium of a large game containing the above three players, if $\Pr[x : 1]$, $\Pr[y : 1] \in \{0,1\}$, then: $\Pr[z : 1] = \Pr[x : 1] \vee \Pr[y : 1]$.

So we obtained an OR gate, and we can similarly obtain AND and NOT gates.

Binary Circuits

Can simulate any boolean circuit with a polymatrix game.



However, cannot enforce that the players will always play pure strategies.

Hence my circuit may not compute something meaningful.

bottom line:

- a reduction restricted to pure strategy equilibria is likely to fail (see also discussion in the last lecture)

- real numbers seem to play a fundamental role in the reduction

Can games do ***real*** *arithmetic?*

What in a Nash equilibrium is capable of storing reals?

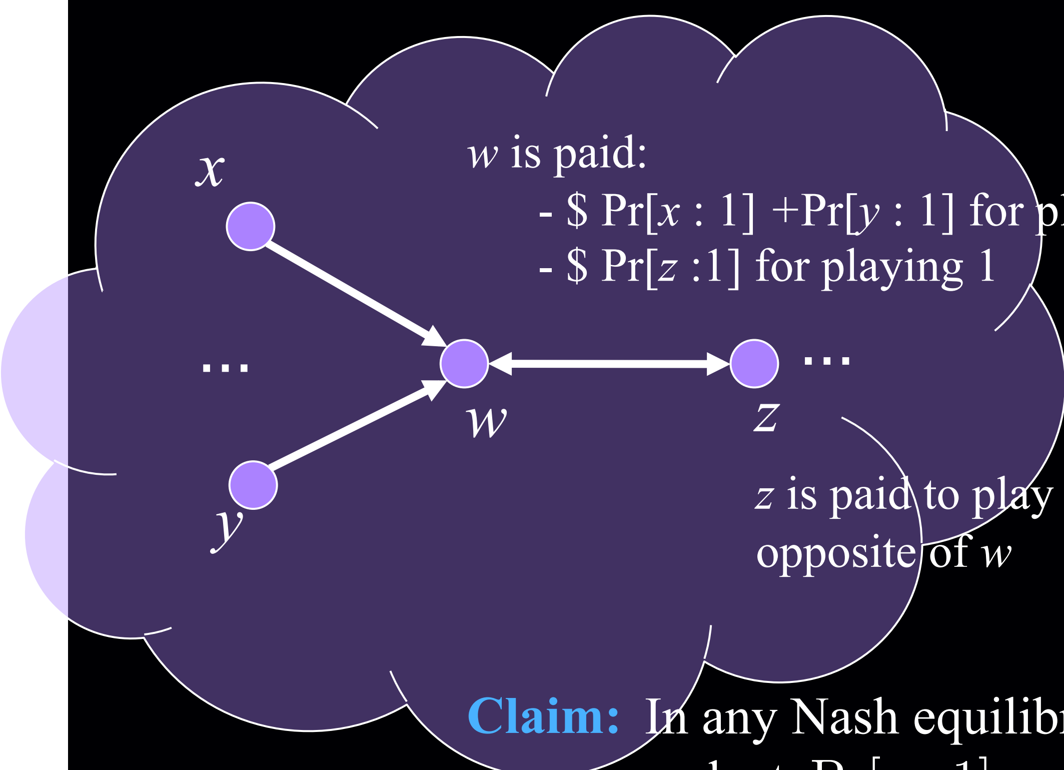
*Games that do **real** arithmetic*

separable

Suppose two strategies per player: {0,1}

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

e.g. *addition game*



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

$$u(z:0) = 0.5$$
$$u(z:1) = 1 - Pr[w:1]$$

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

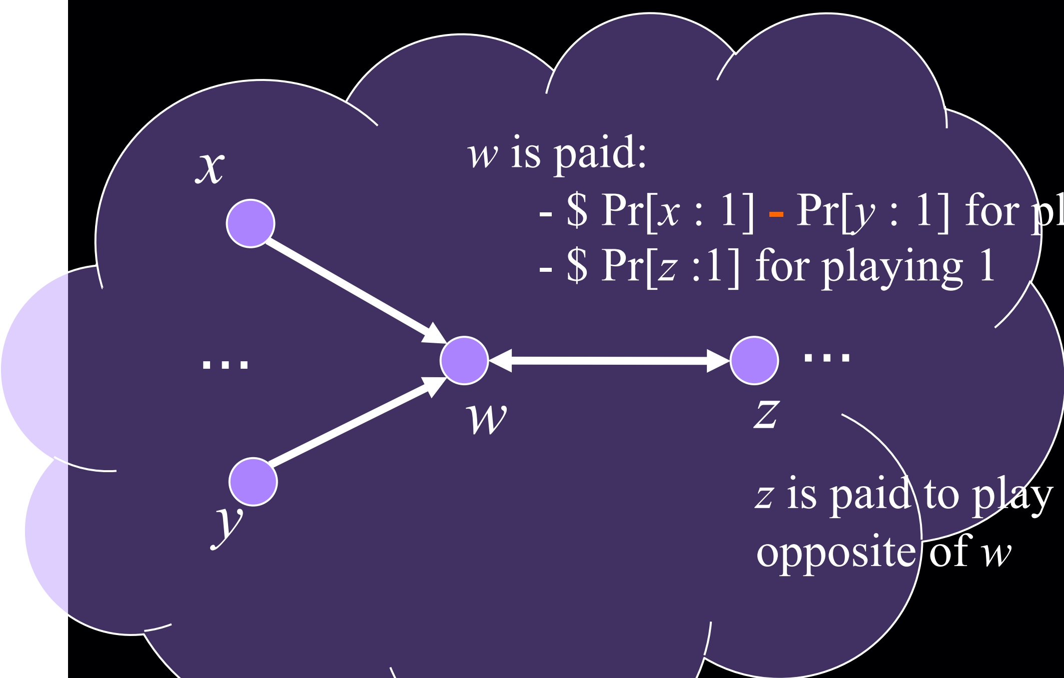
*Games that do **real** arithmetic*

separable

Suppose two strategies per player: {0,1}

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

e.g. *subtraction*



w is paid:

- \$ Pr[*x* : 1] - Pr[*y* : 1] for playing 0
- \$ Pr[*z* :1] for playing 1

z is paid to play the opposite of *w*

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

$$u(z:0) = 0.5$$
$$u(z:1) = 1 - Pr[w:1]$$

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

From now on, use the name of the node and the probability of that node playing 1 interchangeably.

$$x \leftrightsquigarrow \Pr[x:1]$$

*Games that do **real** arithmetic*

copy : $z = x$

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

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

set equal to a constant : $z = \alpha, \text{ for any } \alpha \in [0,1]$

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

separable

---

can also do multiplication $z = x \cdot y$

non separable!

won't be used in our reduction

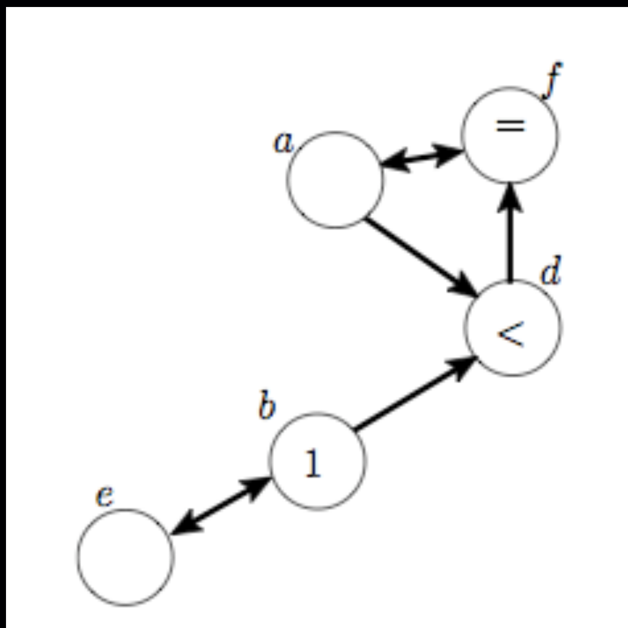
Comparison Gadget

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

brittleness

Comparison Gadget

Impossibility to remove brittleness…



$$d = \begin{cases} 1, & \text{if } a < b \\ 0, & \text{if } a \geq b \end{cases}$$

In any Nash equilibrium:

$$b = 1$$

$$a = d$$

What is a?

$$a = 1 \implies \text{contradiction}$$
$$a < 1 \implies \text{contradiction}$$

Administrativia

Homework:

Scribe notes for Lectures 6, 7 were posted on the website on Friday.

Rule of thumb: Since there will be about 20 lectures in this class, by the end of this week registered students should have collected about 6-7 points in hw problems.

Project:

Groups of 2-3 students (1 is also fine)

Submit a one-page description of the project by next Monday

Preferred: Research Oriented → Study an open problem given in class

→ Come up with your own question (related to the class, or your own area)

Talk to me if you need help

Could also be survey

Our Gates

Constants:

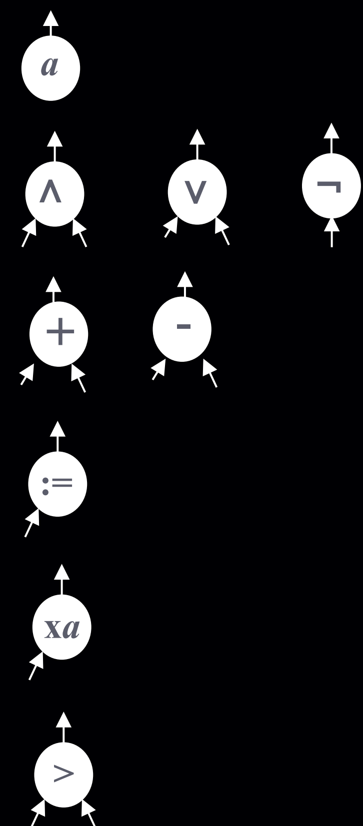
Binary gates:

Linear gates:

Copy gate:

Scale:

Brittle Comparison:



with truncation at 0, 1

any circuit using these gates can be implemented with a polymatrix game

need not be a DAG circuit, i.e. feedback is allowed

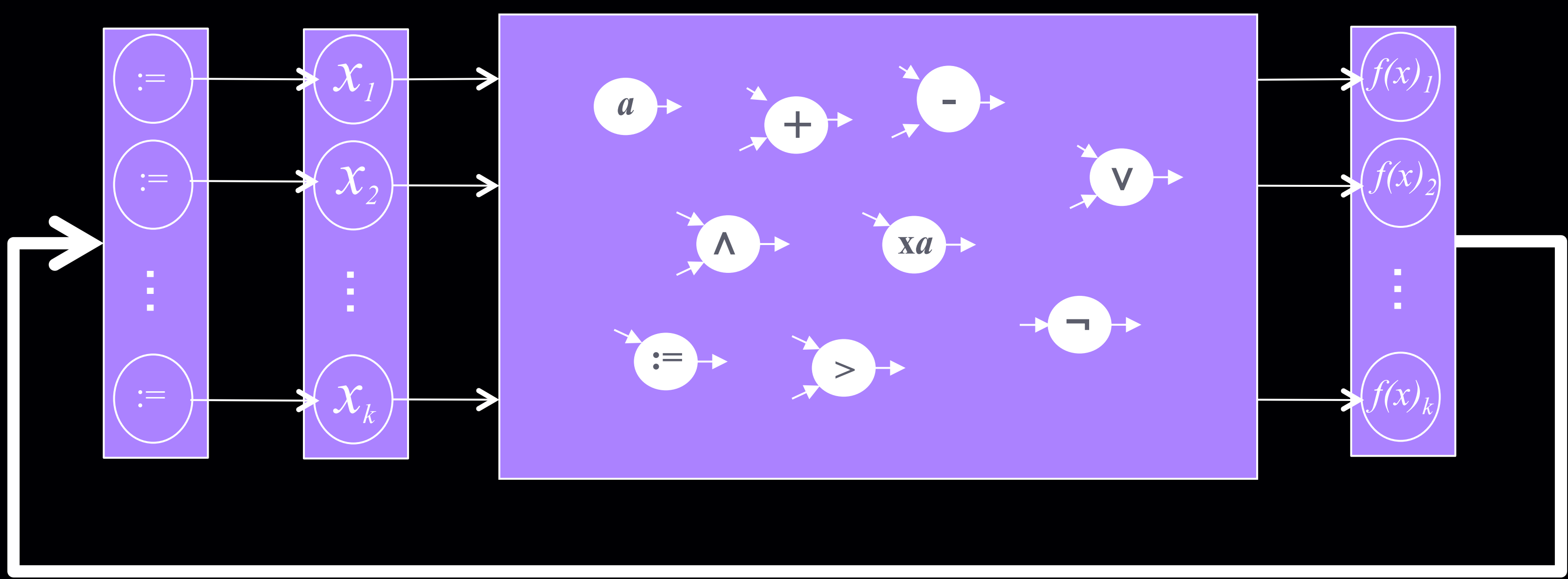
let's call any such circuit a *game-inspired straight-line program*

Fixed Point Computation

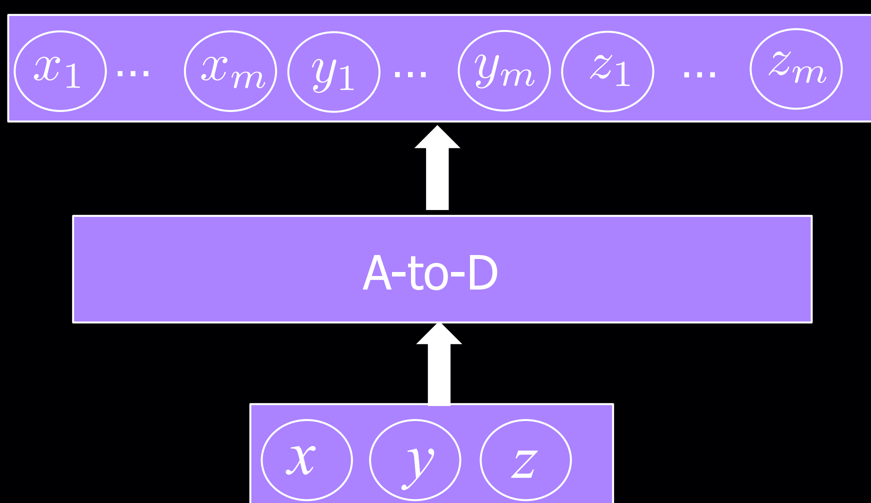
Suppose function $f : [0,1]^k \rightarrow [0,1]^k$ is computed by a game-inspired straight-line program.

➔ Can construct a polymatrix-game whose Nash equilibria are in many-to-one and onto correspondence with the fixed points of *f*.

➔ Can forget about games, and try to reduce PPAD to finding a fixed point of a game-inspired straight-line program.



$\begin{pmatrix} \text{4-displacement} \\ \text{p.w. linear} \end{pmatrix}$ BROUWER → *fixed point of game-inspired straight-line program*



extract m bits from each of x, y, z

three players whose mixed strategies represent a point in $[0,1]^3$

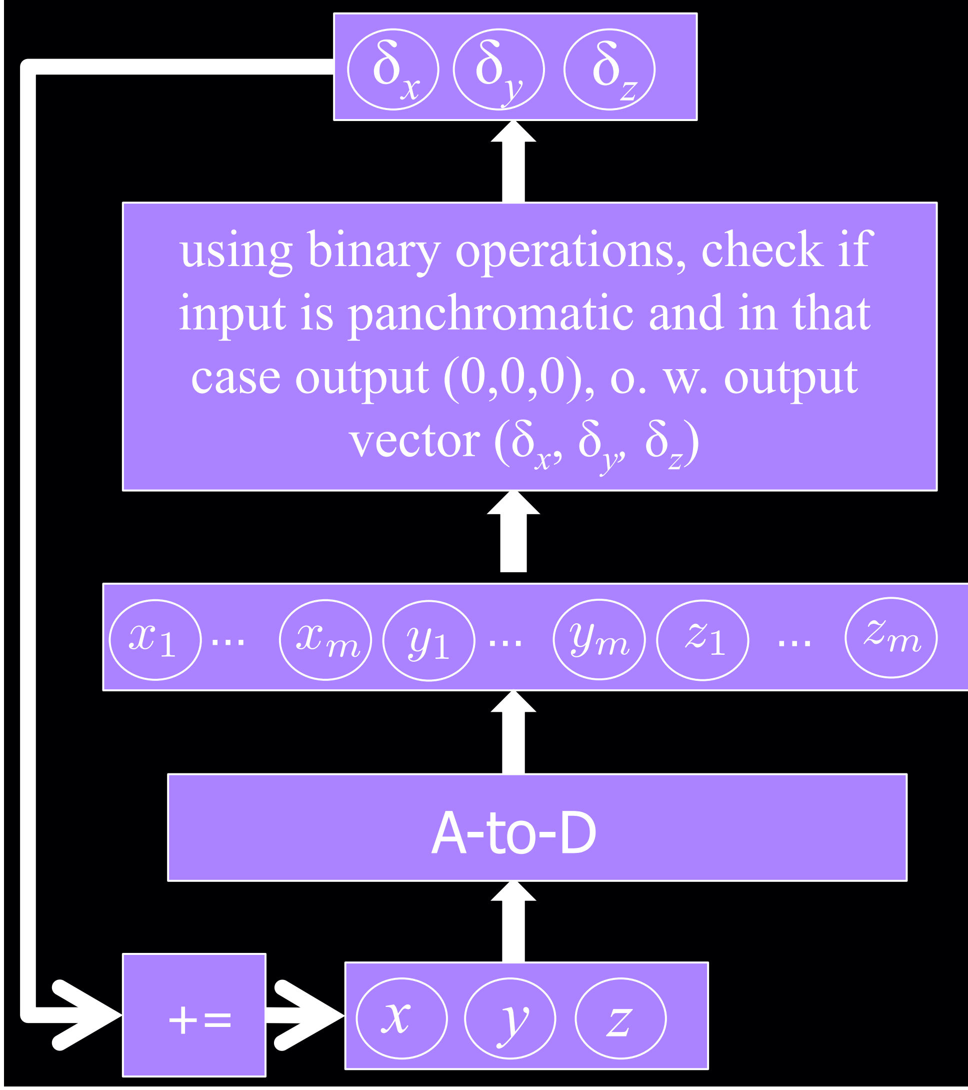
Analog-to-Digital

$$
\begin{aligned}
& v_1 = x; \\
& \text{for } \ i = 1, \ldots, m \text{ do:} \\
& \quad x_i := (2^{-i} \ < \ v_i); \; v_{i+1} := v_i - x_i \cdot 2^{-i}; \\
& \text{similarly for } y \text{ and } z;
\end{aligned}
$$

Can implement the above computation via a game-inspired straight-line program.

The output of the program is always 0/1, except if *x*, *y* or *z* is an integer multiple of 2^{-m}.

(4-displacement p.w. linear) BROUWER → *fixed point of game-inspired straight-line program*



the displacement vector is chosen so that
$(\delta_x, \delta_y, \delta_z) + (x, y, z) \in [0,1]^3$

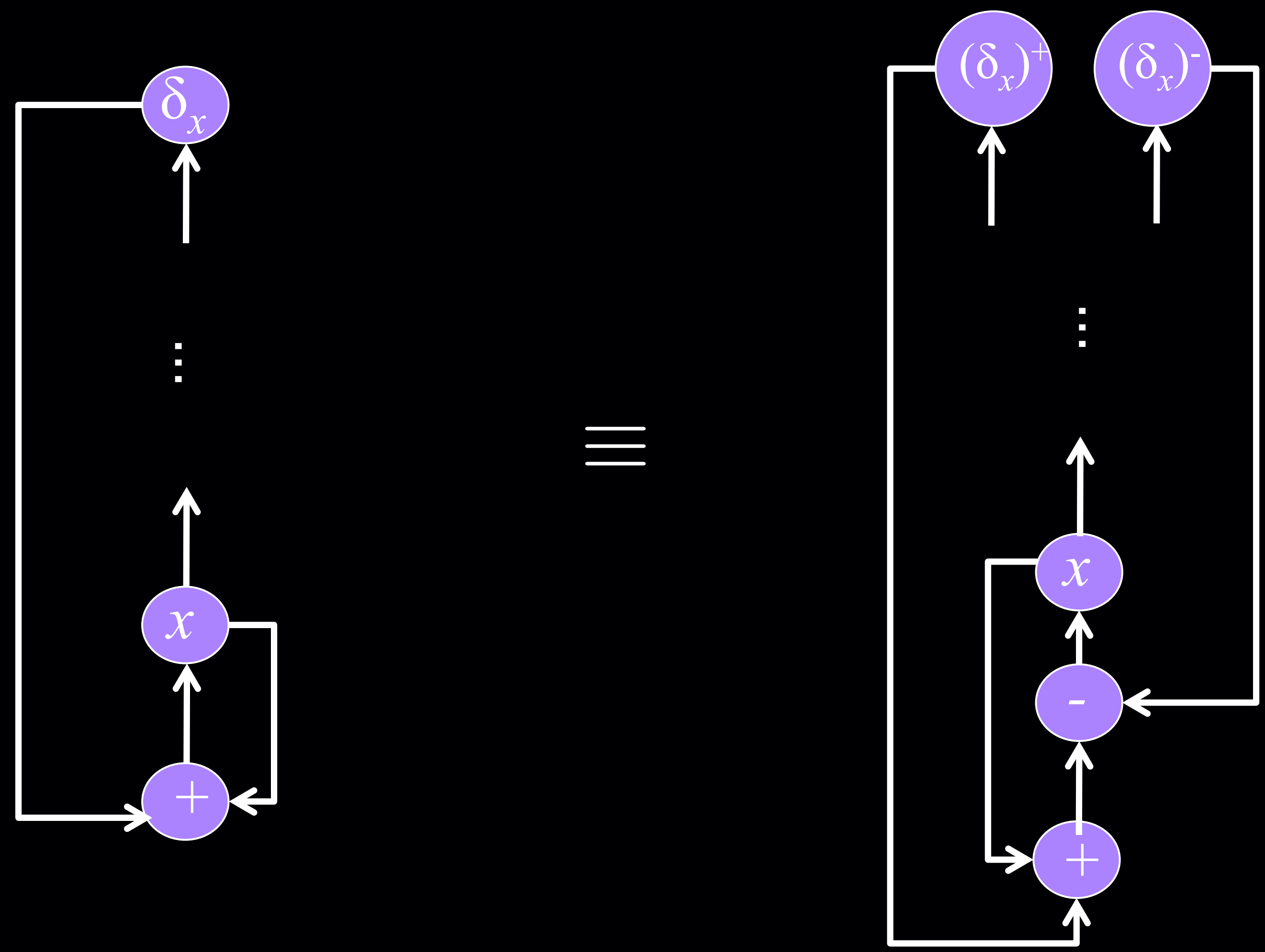
(hopefully) represents a point of the subdivision

extract m bits from each of x, y, z

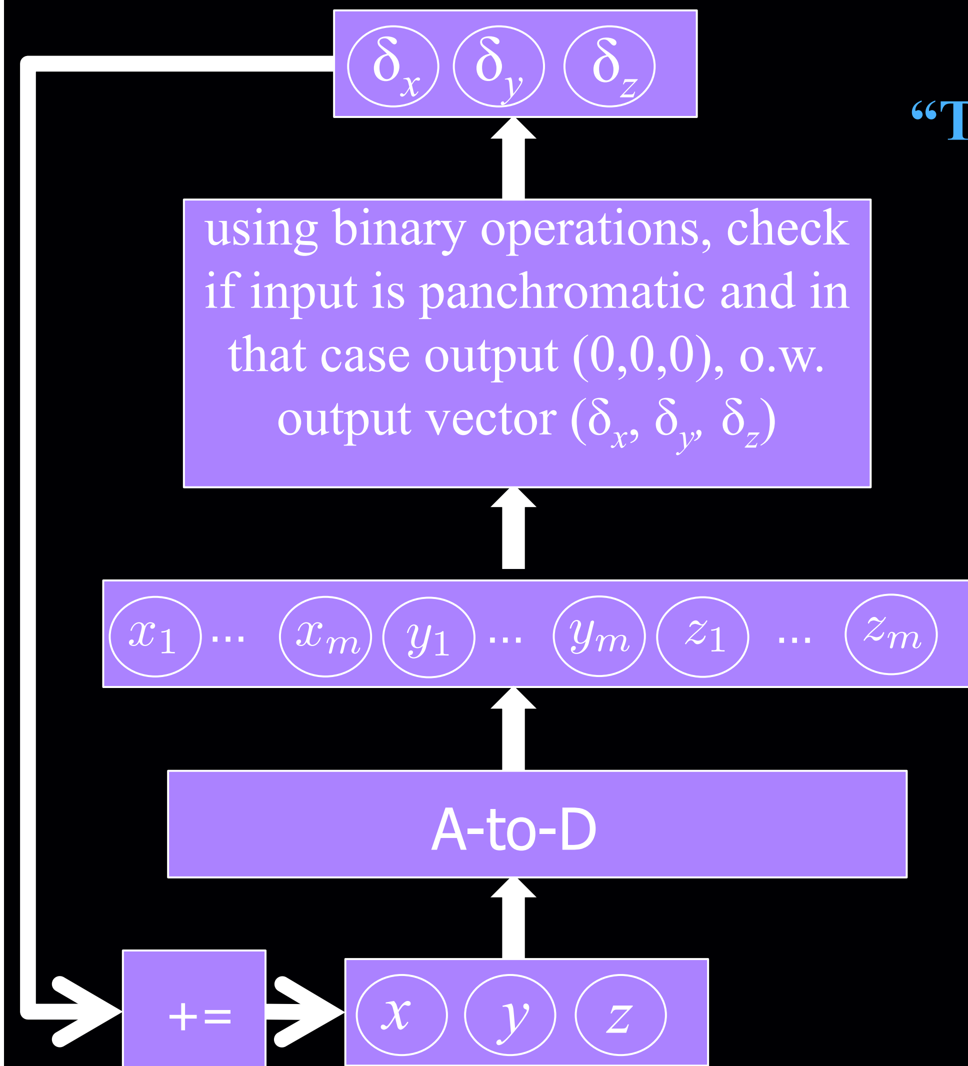
three players whose mixed strategies represent a point in $[0,1]^3$

Add it up

since negative numbers are not allowed



(4-displacement p.w. linear) BROUWER → *fixed point of game-inspired straight-line program*



"Theorem":

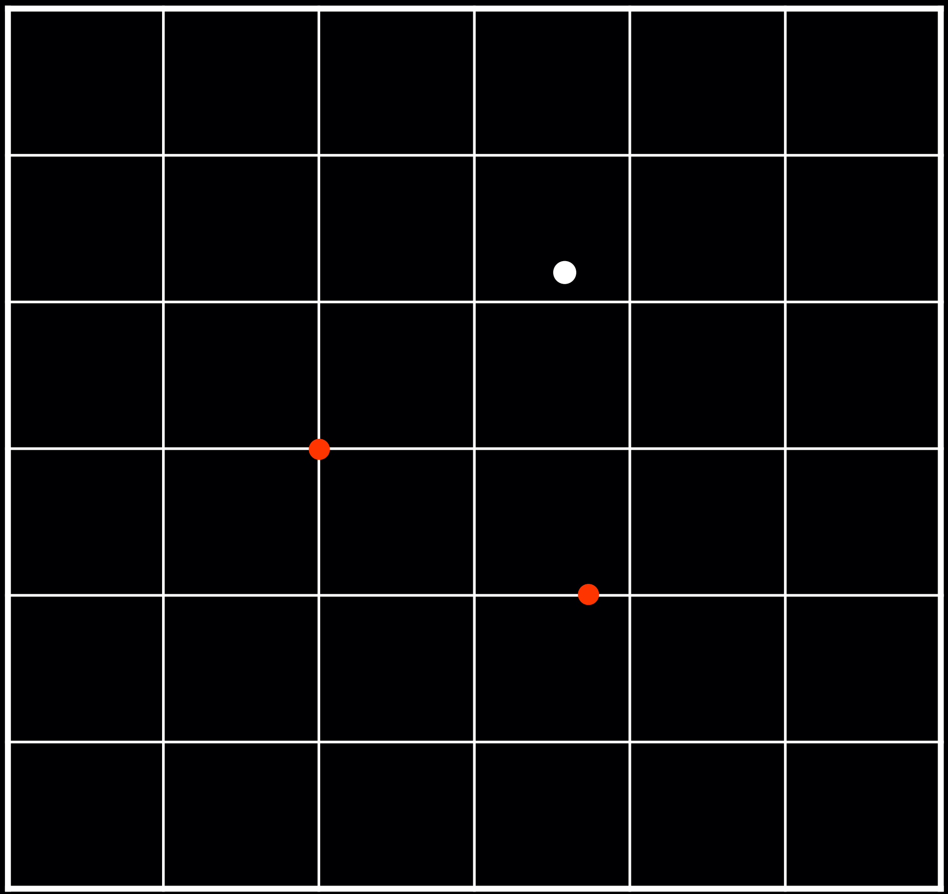
In any fixed point of the circuit shown on the right, the binary description of the point (x, y, z) is panchromatic.

BUT: Brittle comparators don't think so!

this is not necessarily binary

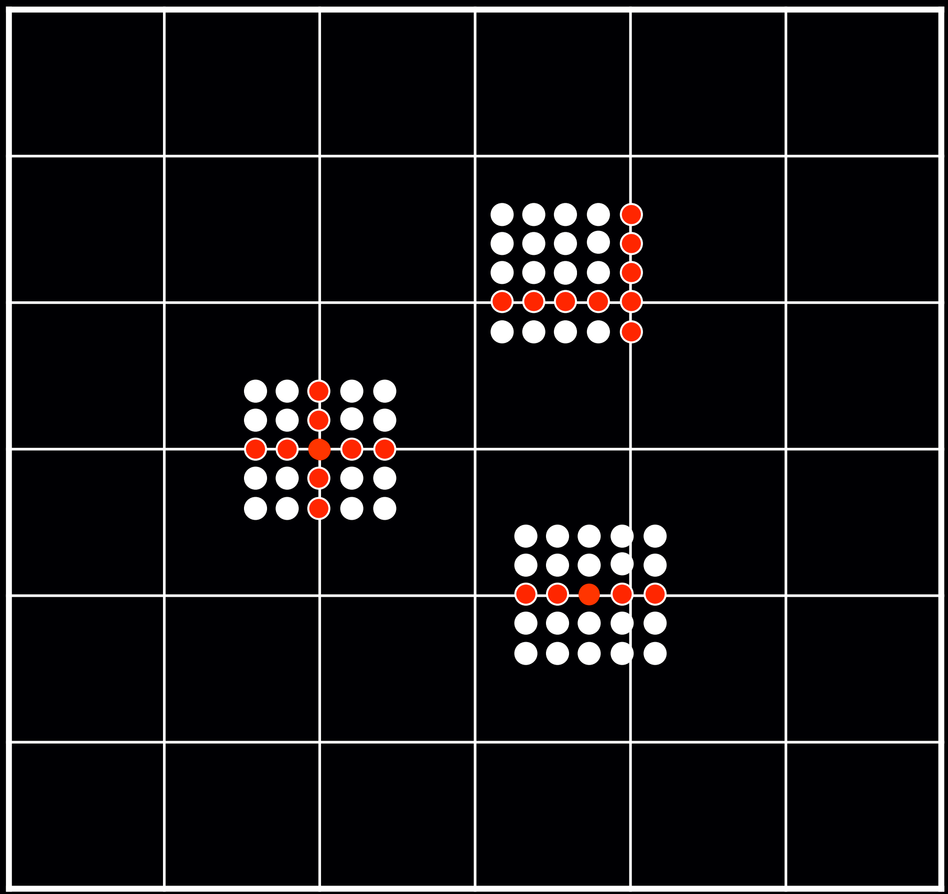
The Final Blow

When did measure-zero sets scare us?



The Final Blow

When did measure-zero sets scare us?



- Create a micro-lattice of copies around the original point (x, y, z):

$$(x + p \cdot 2^{-2m},\ y + q \cdot 2^{-2m},\ z + s \cdot 2^{-2m}),$$
$$-\ell \leq p, q, s \leq \ell$$

- For each copy, extract bits, and compute the displacement of the Brouwer function at the corresponding cubelet, indexed by these bits.

- Compute the average of the displacements found, and add the average to (x, y, z).

Logistics

- There are $M := (2\ell + 1)^3$ copies of the point (x, y, z).

- Out of these copies, at most $3(2\ell + 1)^2$ are broken, i.e. have a coordinate be an integer multiple of 2^{-m}. We cannot control what displacement vectors will result from broken computations.

bad set $\mathcal{B}$

- On the positive side, the displacement vectors computed by at least $(2\ell - 2)(2\ell + 1)^2$ copies correspond to the actual displacement vectors of Brouwer's function.

good set $\mathcal{G}$

- At a fixed point of our circuit, it must be that the (0, 0, 0) displacement vector is added to (x, y, z).

- So the average displacement vector computed by our copies must be (0,0,0).

Theorem: For the appropriate choice of the constant ℓ, even if the set $\mathcal{B}$ "conspires" to output any collection of displacement vectors they want, in order for the average displacement vector to be (0, 0, 0) it must be that among the displacement vectors output by the set $\mathcal{G}$ we encounter all of (1,0,0), (0,1,0), (0,0,1), (-1,-1,-1).

Finishing the Reduction

Theorem: For the appropriate choice of the constant ℓ, even if the set $\mathcal{B}$ "conspires" to output any collection of displacement vectors they want, in order for the average displacement vector to be (0, 0, 0) it must be that among the displacement vectors output by the set $\mathcal{G}$ we encounter all of (1,0,0), (0,1,0), (0,0,1), (-1,-1,-1).

➔ In any fixed point of our circuit, (x, y, z) is in the proximity of a point (x^*, y^*, z^*) of the subdivision surrounded by all four displacements. This point can be recovered in polynomial time given (x, y, z).

➔ in any Nash equilibrium of the polymatrix game corresponding to our circuit the mixed strategies of the players x, y, z define a point located in the proximity of a point (x^*, y^*, z^*) of the subdivision surrounded by all four displacements. This point can be recovered in polynomial time given (x, y, z).

$\Longrightarrow$ (exact) POLYMATRIX NASH is PPAD-complete

Finishing the Reduction

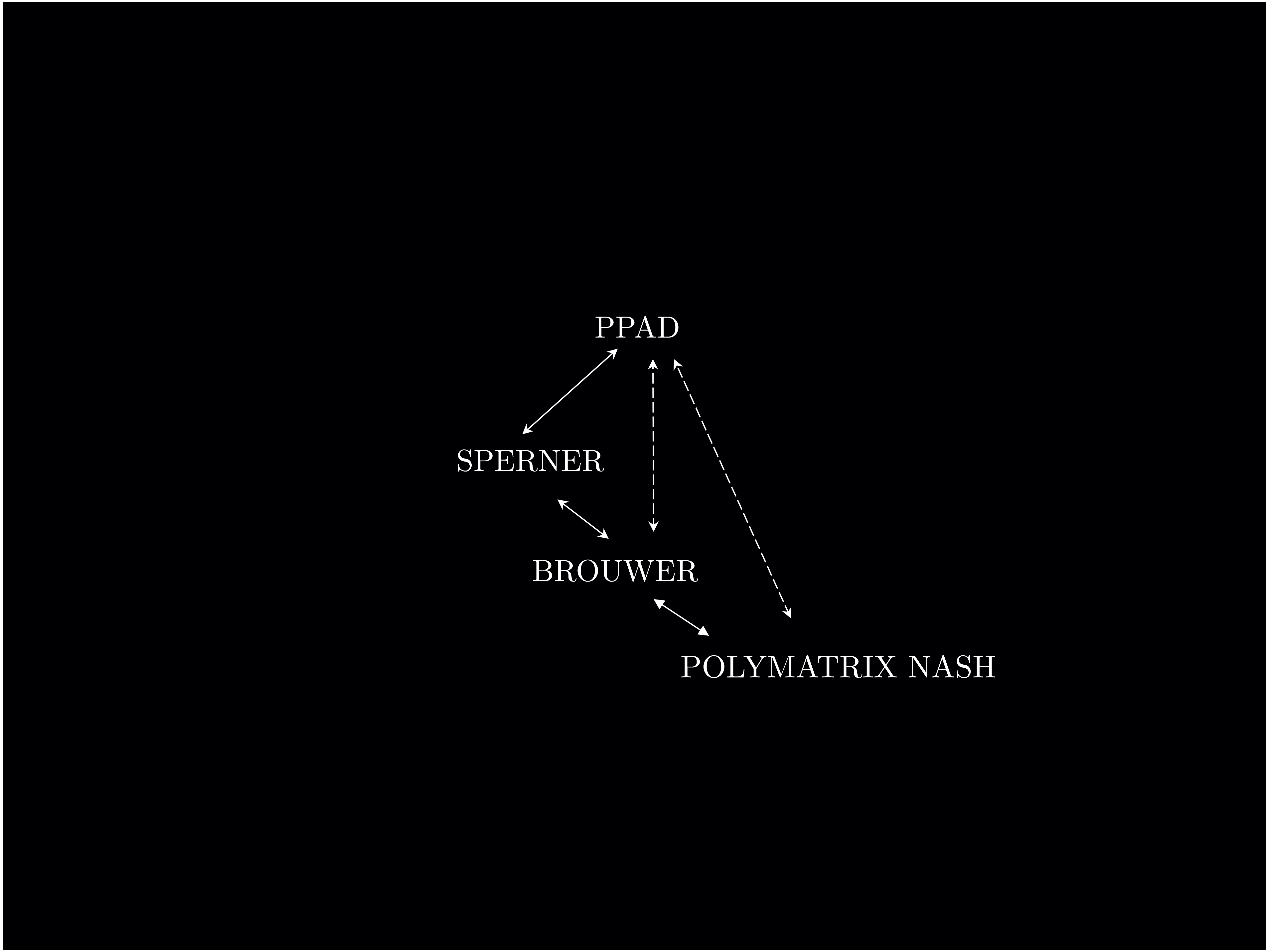
Theorem: Given a polymatrix game $\mathcal{G}$ there exists ϵ^* such that:

1. $|\epsilon^*| = \text{poly}(|\mathcal{G}|)$
2. given a ϵ^*-Nash equilibrium of $\mathcal{G}$ we can find in polynomial time an exact Nash equilibrium of $\mathcal{G}$.

Proof: 2 points

$$\Longrightarrow \quad \text{(exact) POLYMATRIX NASH} \equiv \text{POLYMATRIX NASH}$$

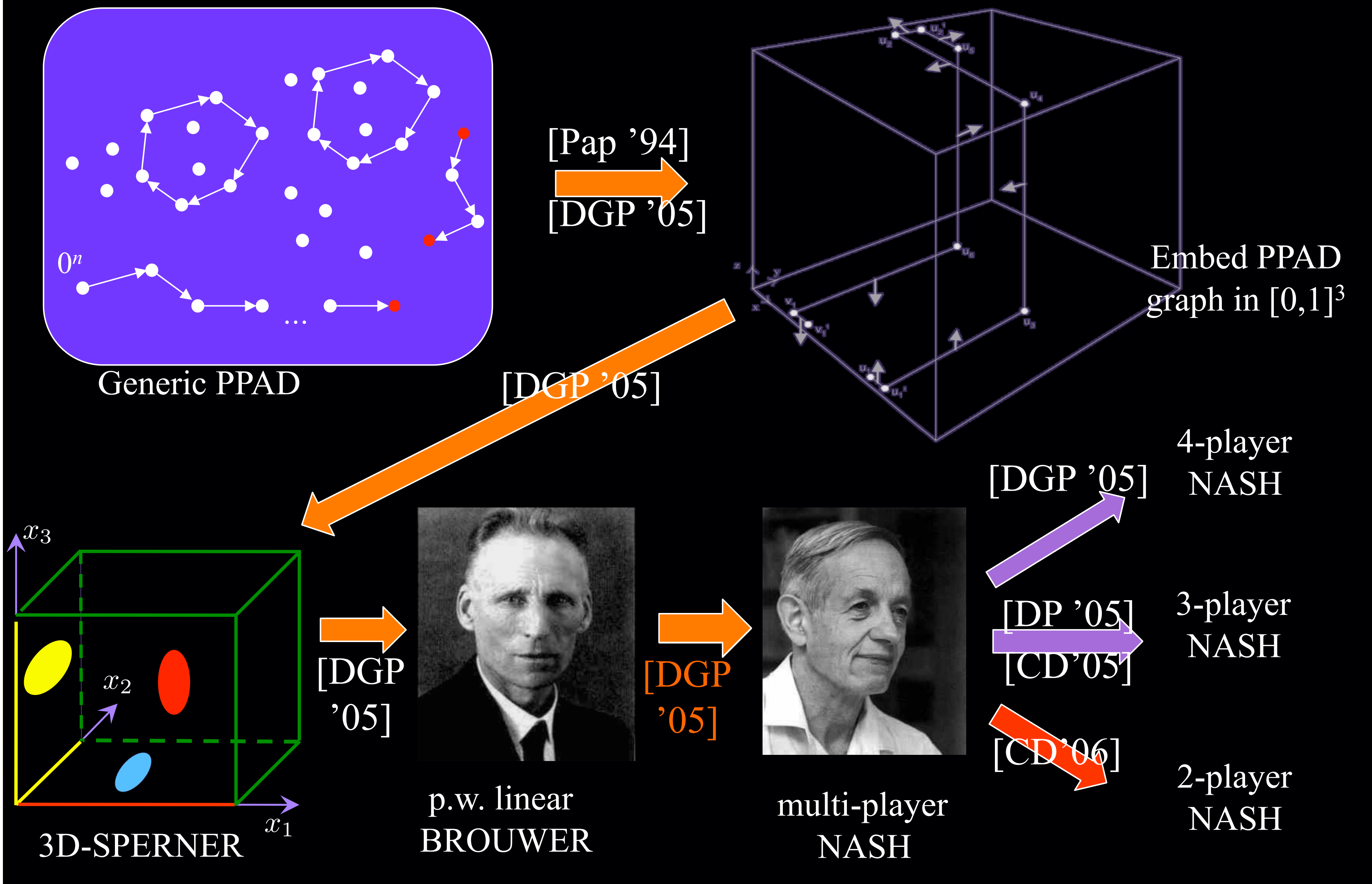
$$\Longrightarrow \quad \text{POLYMATRIX NASH is PPAD-complete}$$


PPAD
SPERNER
BROUWER
POLYMATRIX NASH

This Lecture

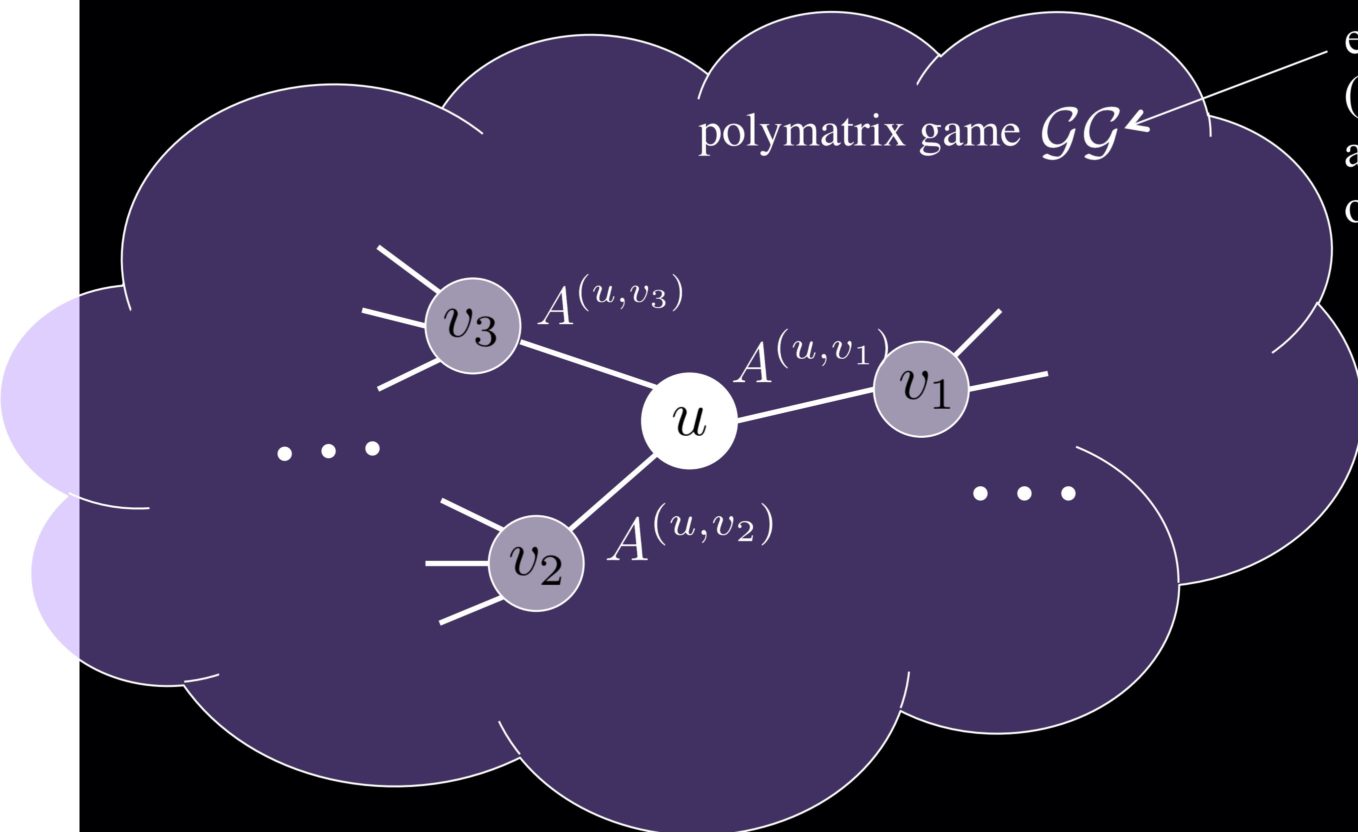
DGP = Daskalakis, Goldberg, Papadimitriou

CD = Chen, Deng



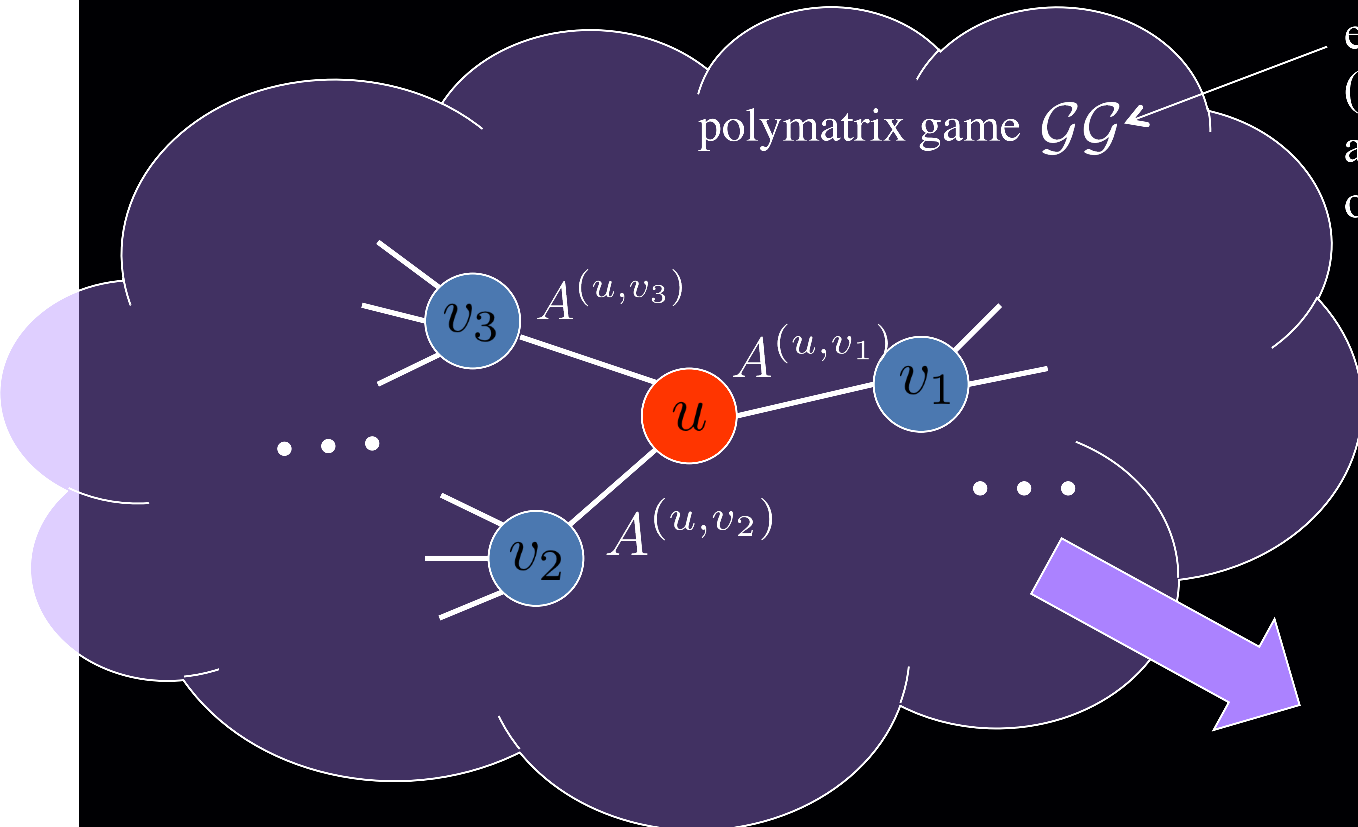
Reducing to 2 players

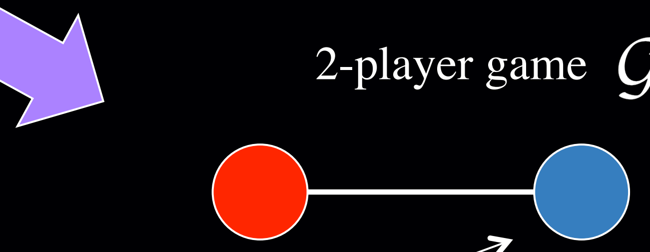
can assume bipartite, by turning every gadget into a bipartite game (inputs&output are on one side and "middle player" is on the other



Reducing to 2 players

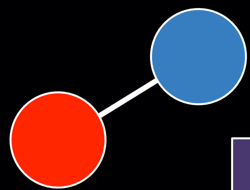
can assume bipartite, by turning every gadget into a bipartite game (inputs&output are on one side and “middle player” is on the other

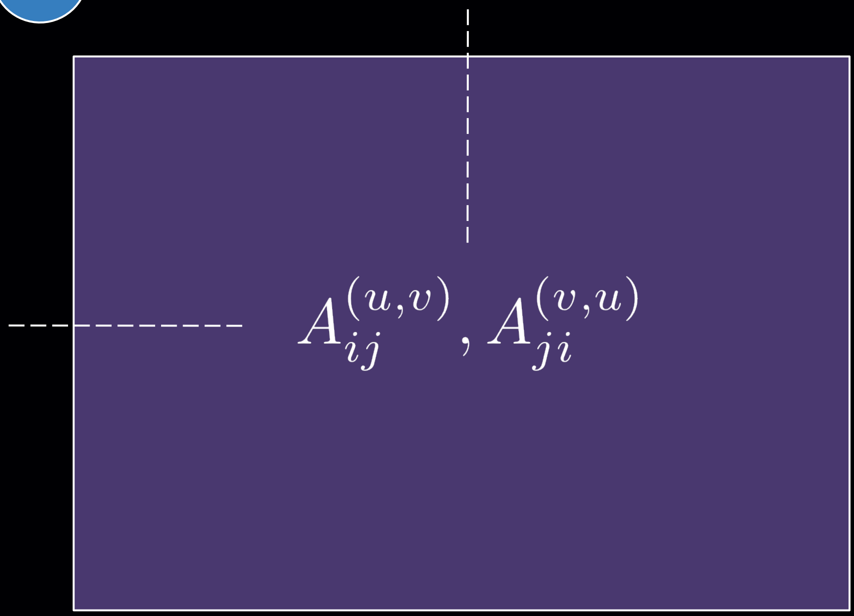


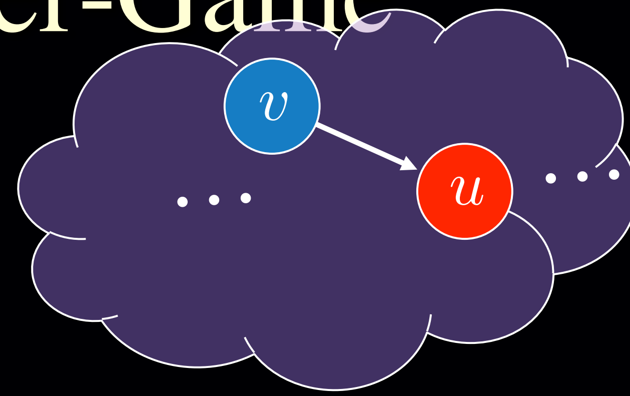


red lawyer represents red nodes, while
blue lawyer represents blue nodes

Payoffs of the Lawyer-Game







But why would a lawyer play every node he represents?

- *wishful thinking*: if (x , y) is a Nash equilibrium of the lawyer-game, then the marginal distributions that x assigns to the strategies of the red nodes and the marginals that y assigns to the blue nodes, comprise a Nash equilibrium.

Enforcing Fairness

- The lawyers play on the side a high-stakes game.

- W.l.o.g. assume that each lawyer represents n clients. Name these clients 1,…,n.

- Payoffs of the high-stakes game:

||

M

Suppose the red lawyer plays any strategy of client j, and blue lawyer plays any strategy of client k, then

If $j \neq k$, then both players get 0.

If $j = k$, then red lawyer gets +M, while blue lawyer gets –M.

Enforcing Fairness

Claim: The unique Nash equilibrium of the high-stakes lawyer game is for both lawyers to play uniformly over their clients.

Proof: 1/2 point

Enforcing Fairness

$(v : j)$

$(u : i)$ $A_{ij}^{(u,v)}, A_{ji}^{(v,u)}$

+

payoff table addition

$$M > 2n \cdot u_{\max}$$

$u_{\max}$: maximum absolute value in payoff tables

high stakes game

M,-M	0,0	0,0
0,0	M,-M	0, 0
0, 0	0 , 0	M,-M

M =

Analyzing the Lawyer Game

- when it comes to distributing the total probability mass among the different nodes of $\mathcal{GG}$, essentially only the high-stakes game is relevant to the lawyers…

Lemma 1: if (x, y) is an equilibrium of the lawyer game, for all u, v :

$$x_u = \frac{1}{n} \cdot \left(1 \pm \frac{2u_{\max}n^2}{M}\right) \qquad y_v = \frac{1}{n} \cdot \left(1 \pm \frac{2u_{\max}n^2}{M}\right)$$

Proof: 1.5 points

total probability mass assigned by lawyers on nodes u, v respectively

- when it comes to distributing the probability mass x_u among the different strategies of node u, only the payoffs of the game $\mathcal{GG}$ are relevant…

Lemma 2: The payoff difference for the red lawyer from strategies $(u : i)$ and $(u : j)$ is

$$\sum_v \sum_\ell \left(A_{i,\ell}^{(u,v)} - A_{j,\ell}^{(u,v)}\right) \cdot y_{v:\ell}$$

Analyzing the Lawyer Game (cont.)

Lemma 2 ➔ if $x_{u:i} > 0$, then for all j:

$$\sum_v \sum_\ell \left(A^{(u,v)}_{i,\ell} - A^{(u,v)}_{j,\ell} \right) \cdot y_{v:\ell} \geq 0$$

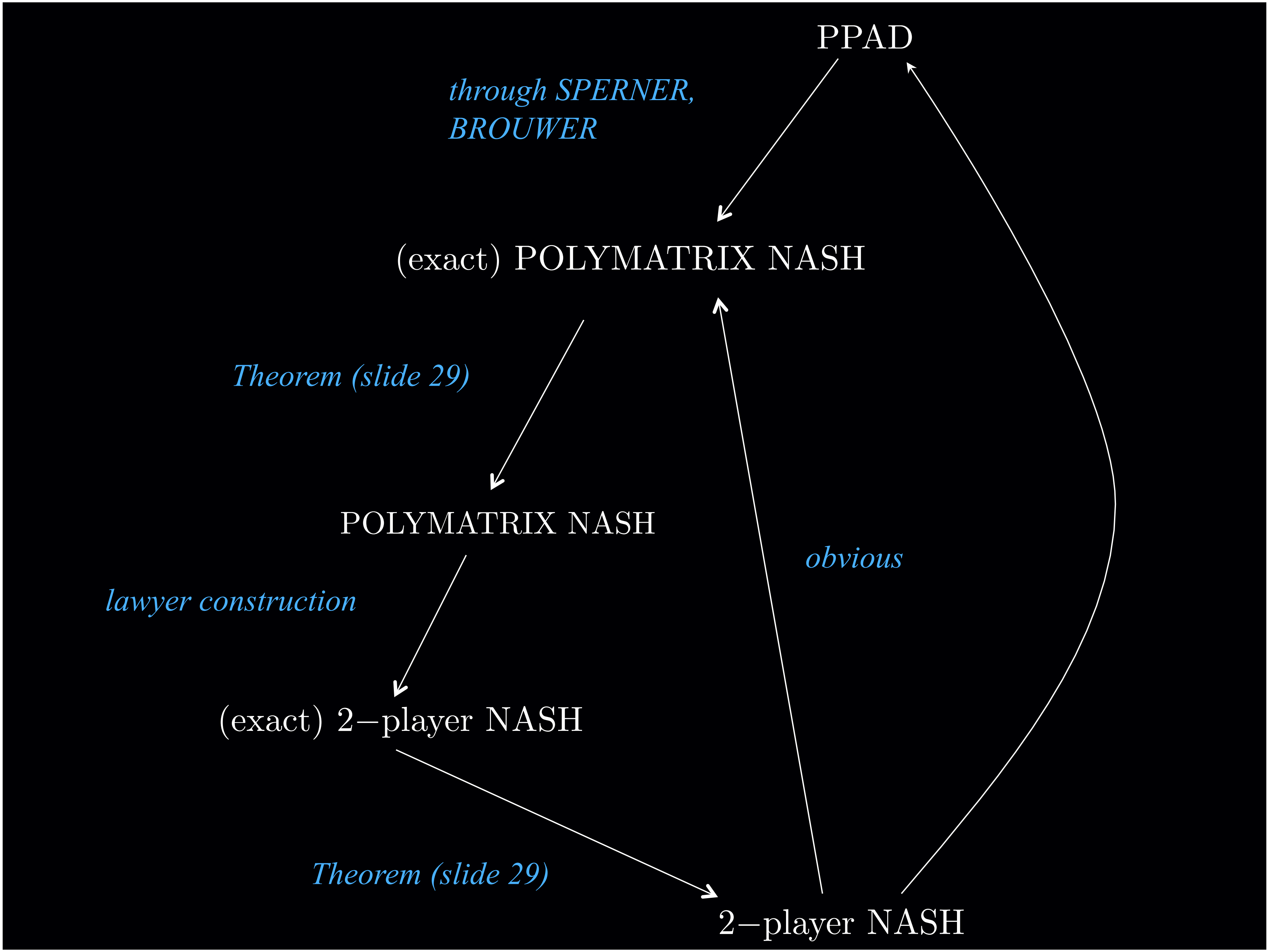
- define $\hat{x}_u(i) := \frac{x_{u:i}}{x_u}$ and $\hat{y}_v(j) := \frac{y_{v:j}}{y_v}$ (marginals given by lawyers to different nodes)

Observation: if we had $x_u = 1/n$, for all u, and $y_v = 1/n$, for all v, then

$\{\{\hat{x}_u\}_u, \{\hat{y}_v\}_v\}$ would be a Nash equilibrium.

- the $\pm \frac{2u_{\max}n}{M}$ deviation from uniformity results in an approximate Nash equilibrium of the polymatrix game.

- if M is large, can correct it to an exact Nash equilibrium of the polymatrix game, appealing to Theorem of Slide 29. $\square$


PPAD
through SPERNER,
BROUWER
(exact) POLYMATRIX NASH
Theorem (slide 29)
POLYMATRIX NASH
obvious
lawyer construction
(exact) 2−player NASH
Theorem (slide 29)
2−player NASH