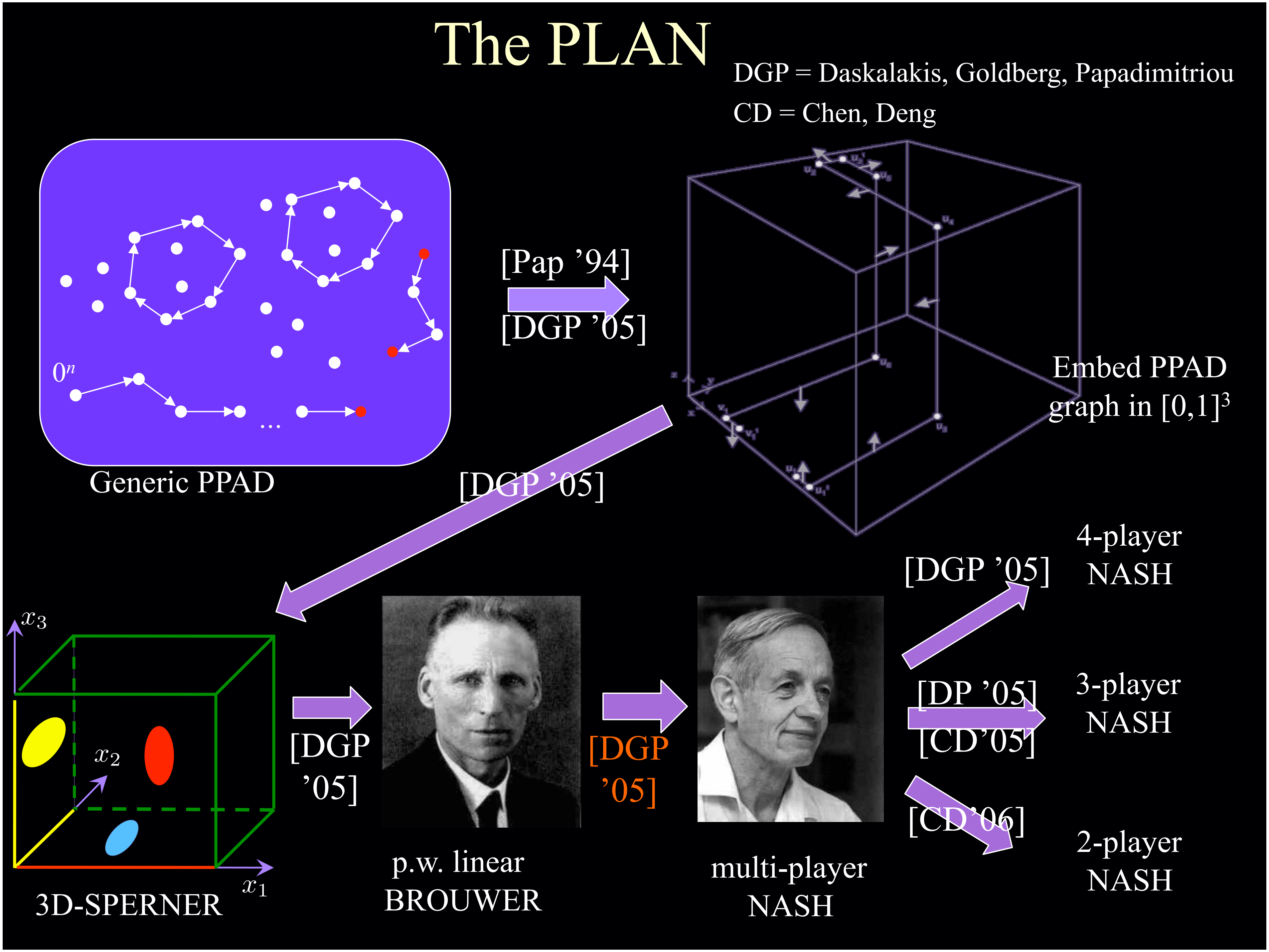
6.896: Topics in Algorithmic Game Theory

Lecture 9

Constantinos Daskalakis

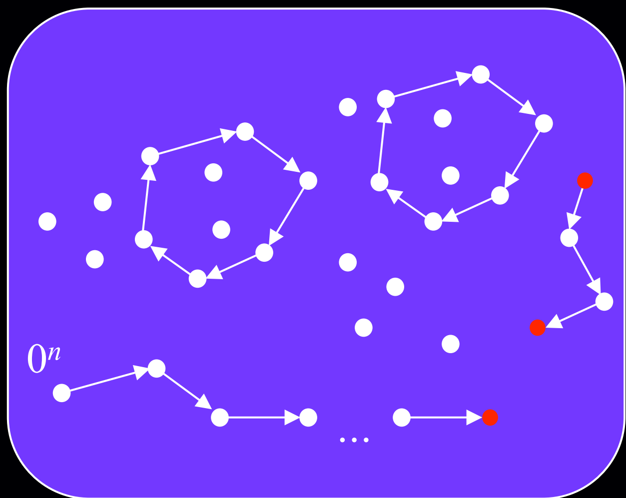
Last Time…


The PLAN
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]
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

Last Lecture

DGP = Daskalakis, Goldberg, Papadimitriou

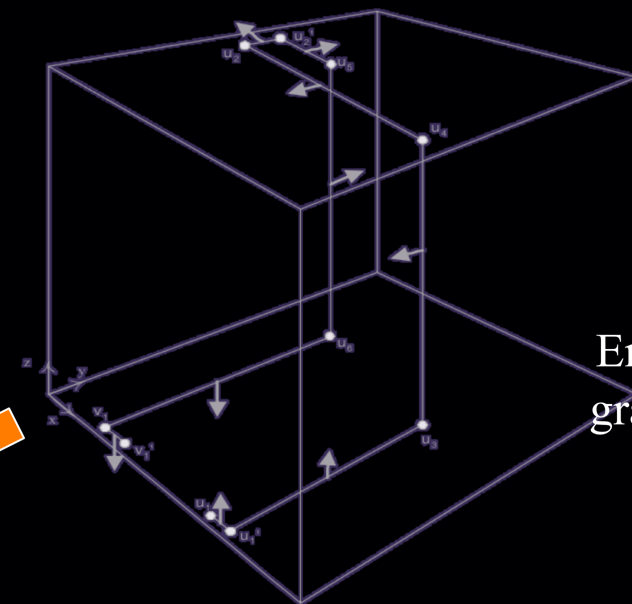
CD = Chen, Deng



Generic PPAD

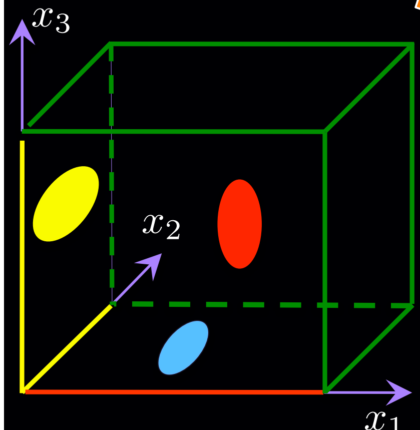
[Pap '94]

[DGP '05]



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

[DGP '05]

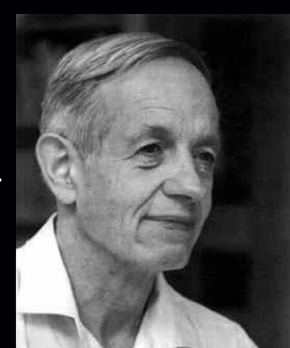


3D-SPERNER

[DGP '05]

p.w. linear BROUWER

[DGP '05]



multi-player NASH

[DGP '05]

4-player NASH

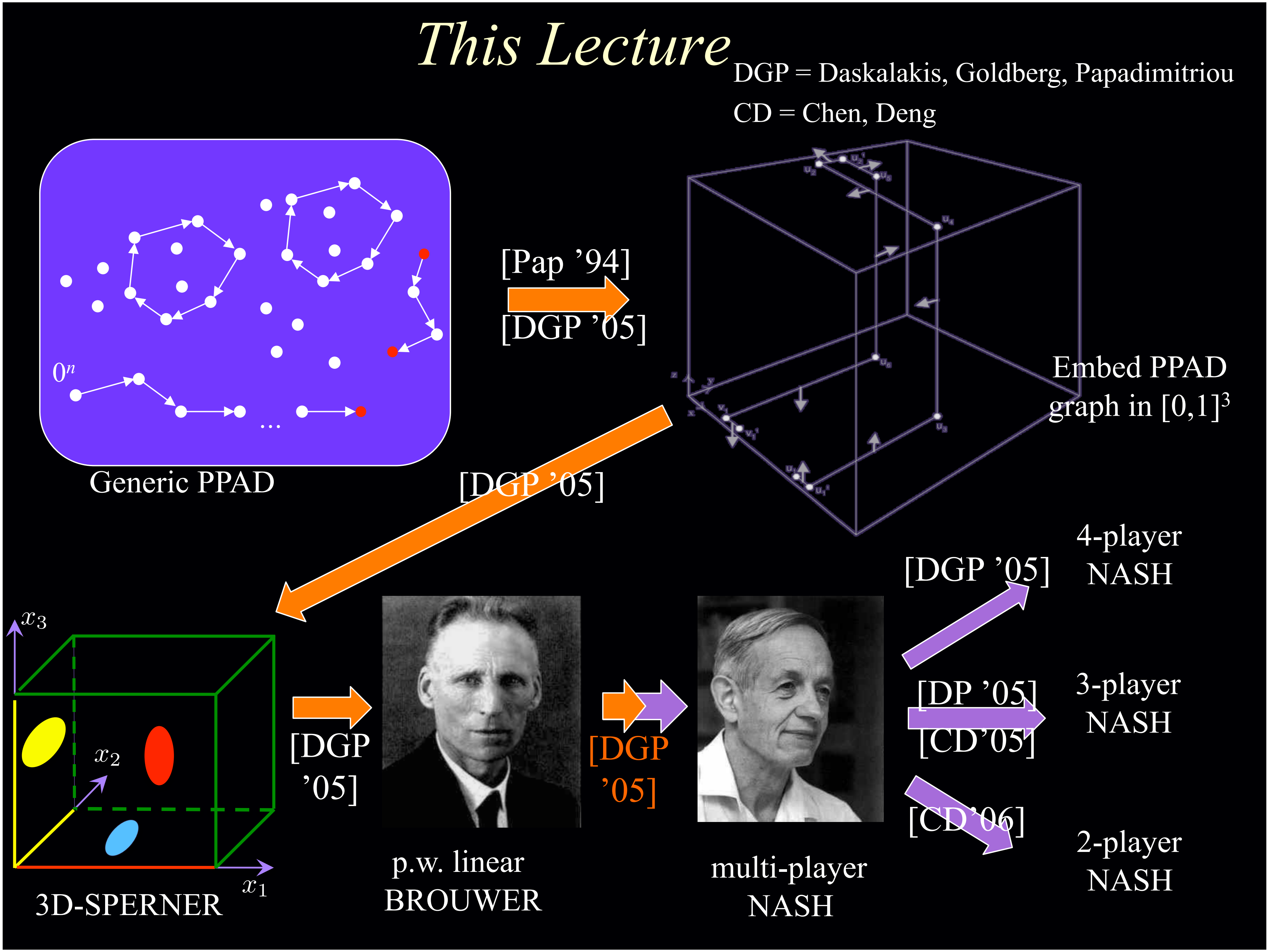
[DP '05]

[CD'05]

3-player NASH

[CD'06]

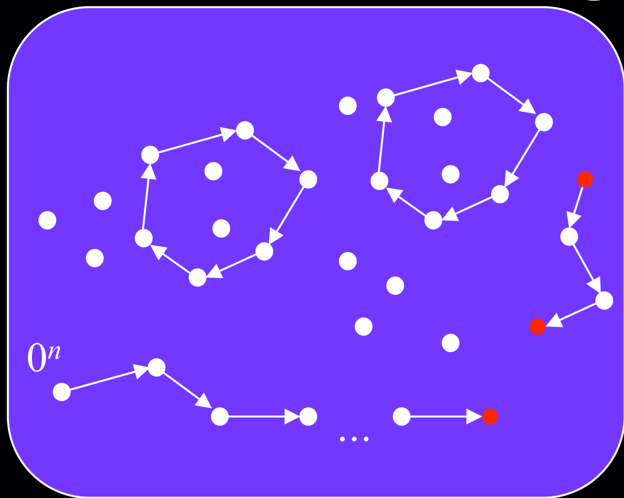
2-player NASH


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

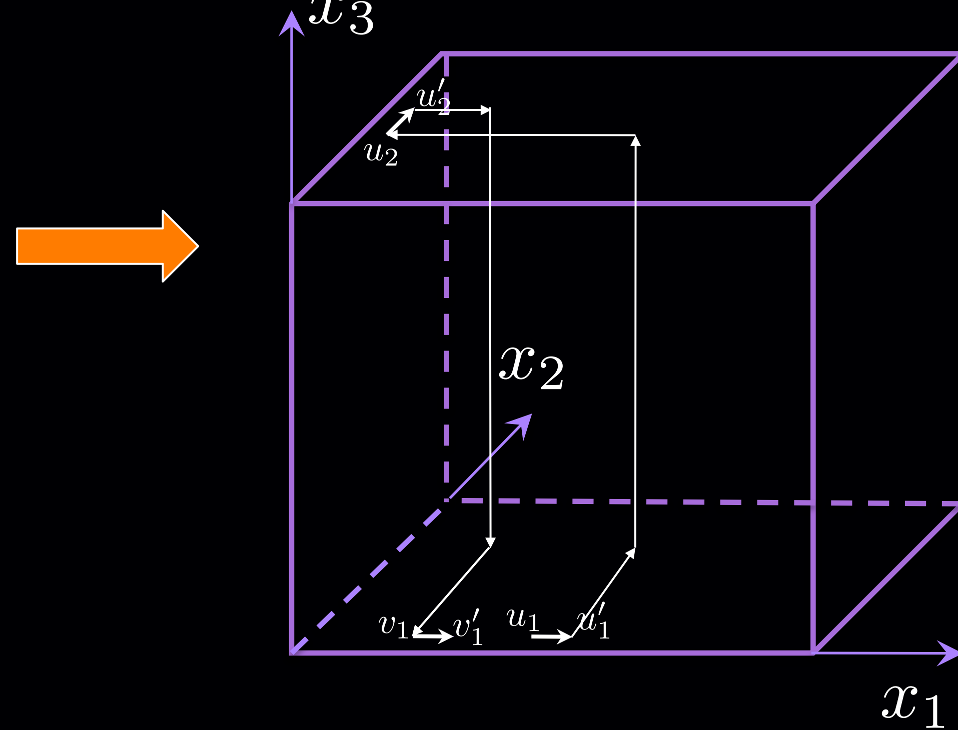
Review of Last Lecture…

PPAD-completeness of SPERNER

Embedding of PPAD graph into $[0,1]^3$



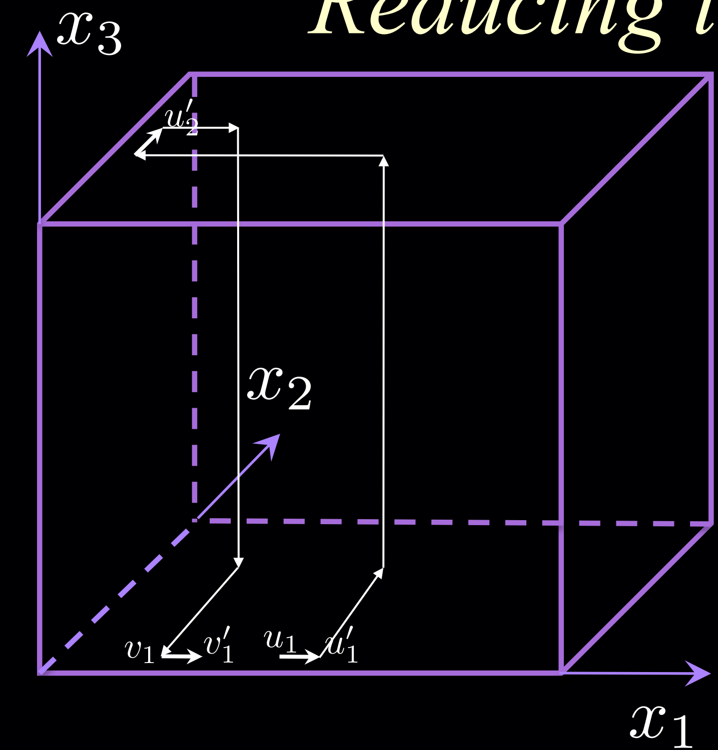
Generic PPAD

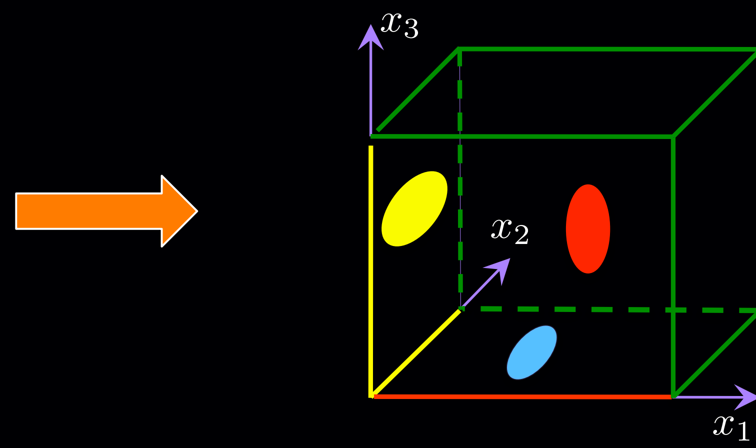


Non-Isolated Node → *pair of segments (the main and the auxiliary segment); connected by an orthonormal line.*

Edge between u and v → *orthonormal path connecting the end of the auxiliary segment of u with the beginning of main segment of v*

Reducing to 3-d Sperner

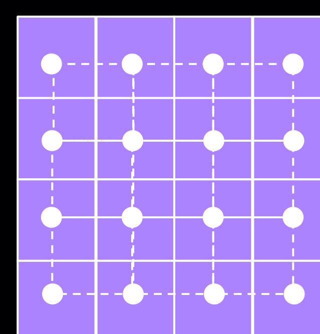




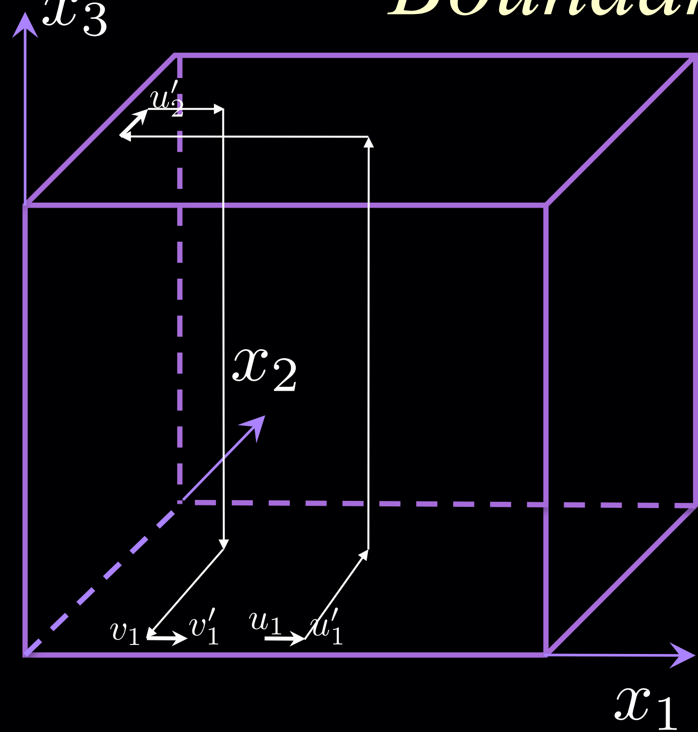
3-d SPERNER

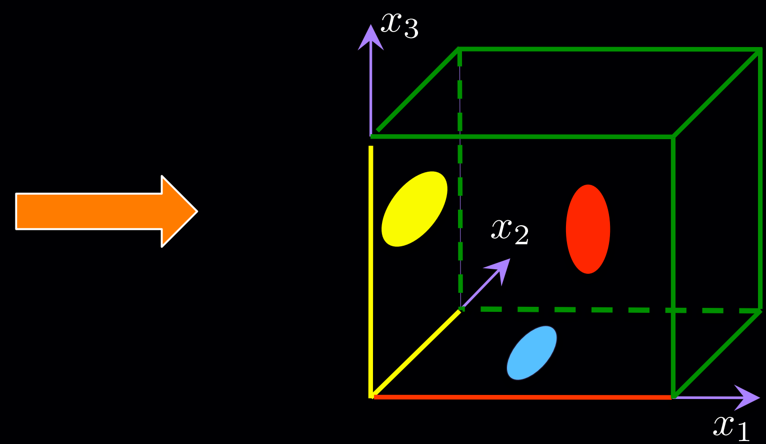
Instead of coloring vertices of the triangulation (the points of the cube whose coordinates are integer multiples of 2^{-m}), color the centers of the cubelets; i.e. work with the dual graph.

K_{ijk} : center of cubelet whose least significant corner has coordinates $(i, j, k) \cdot 2^{-m}$



Boundary Coloring



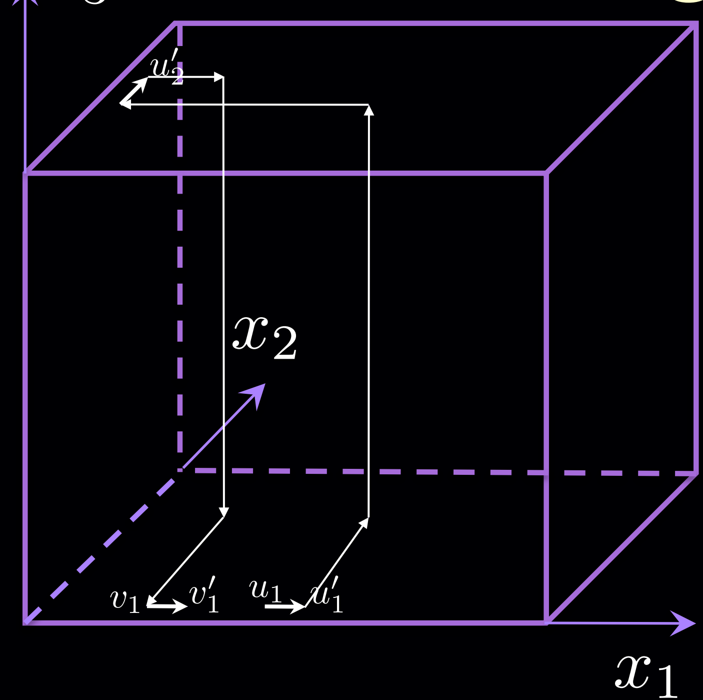


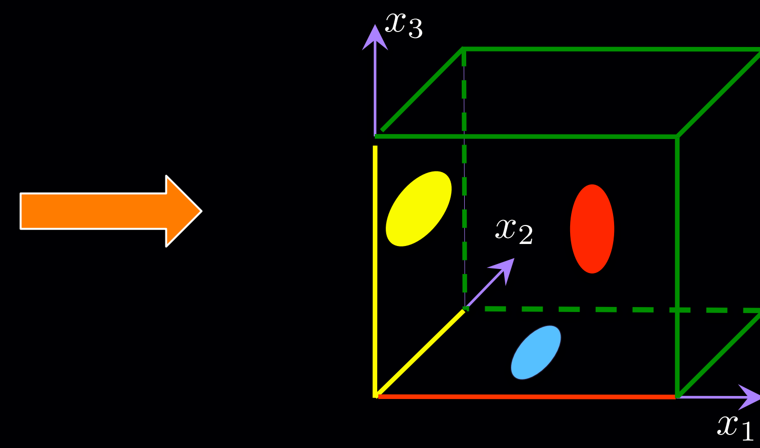
$$\left.\begin{aligned}
&K_{ijk} \leftarrow 0, \quad \text{if any of } i,\, j,\, k \text{ is } 2^m - 1\\
&K_{ijk} \leftarrow 1, \quad \text{if } i = 0\\
&K_{ijk} \leftarrow 2, \quad \text{if } j = 0\\
&K_{ijk} \leftarrow 3, \quad \text{if } k = 0
\end{aligned}\right\}$$

legal coloring for the dual graph (on the centers of cubelets)

N.B.: this coloring is not the envelope coloring we used earlier; also color names are permuted

Coloring of the Rest

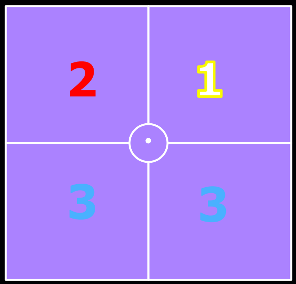




Rest of the coloring: *All cubelets get color* ***0****, unless they touch line L.*

The cubelets surrounding line L at any given point are colored with colors ***1****, 2, 3 in a way that "protects" the line from touching color 0.*

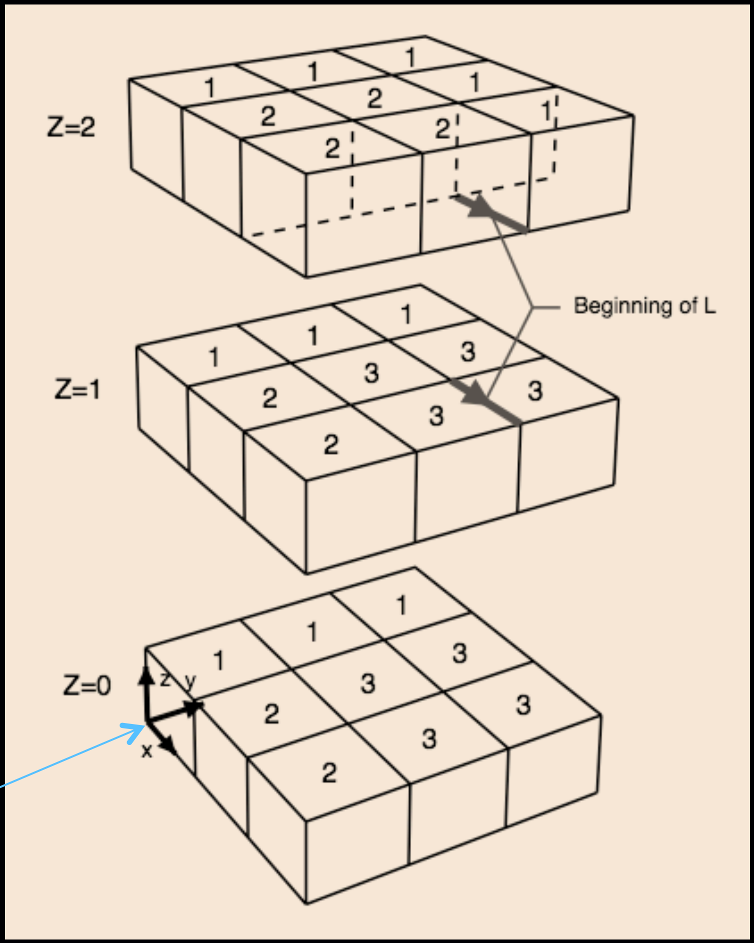
Coloring around L



*colors **1**, 2, 3 are placed in a clockwise arrangement for an observer who is walking on L*

two out of four cubelets are colored 3, one is colored 1 and the other is colored 2

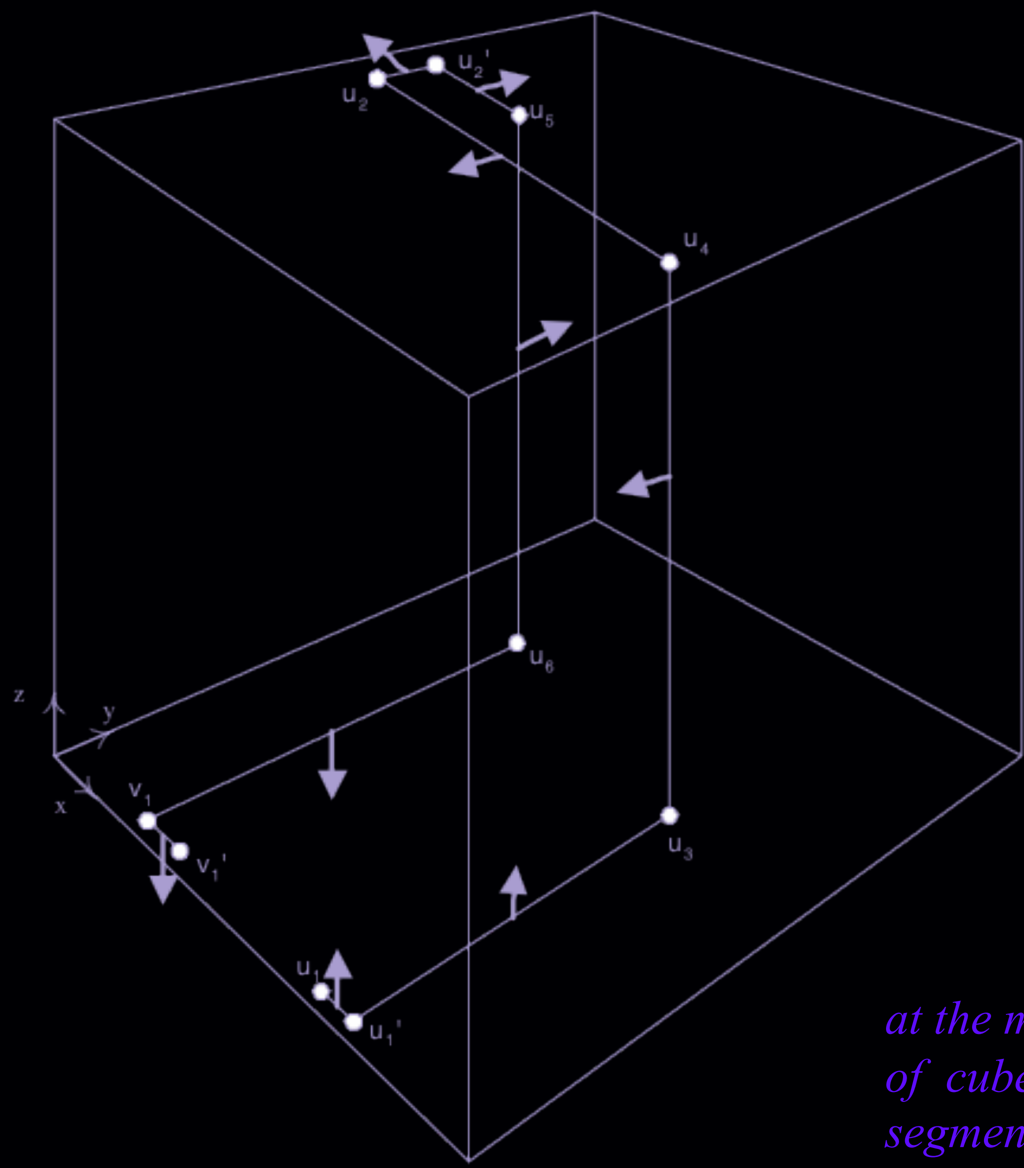
The Beginning of L at 0^n



$(0, 0, 0)$

notice that given the coloring of the cubelets around the beginning of L (on the left), there is no point of the subdivision in the proximity of these cubelets surrounded by all four colors…

Color Twisting



out of the four cubelets around L which two are colored with color 3 ?

- in the figure on the left, the arrow points to the direction in which the two cubelets colored 3 lie

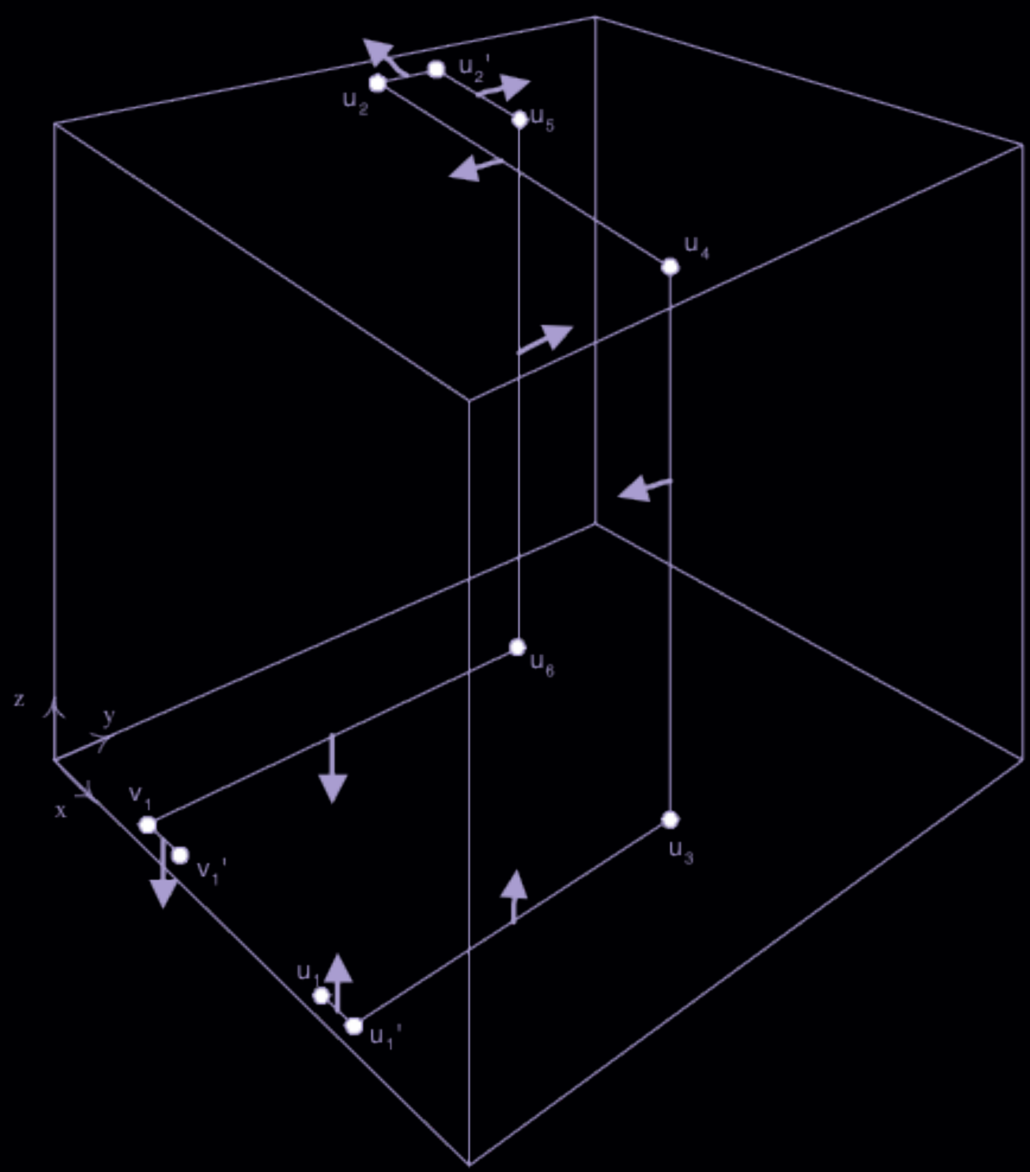
- observe also the way the twists of L affect the location of these cubelets with respect to L

IMPORTANT directionality issue:

the picture on the left shows the evolution of the location of the pair of colored 3 cubelets along the subset of L corresponding to an edge (u, v) of the PPAD graph…

at the main segment corresponding to u the pair of cubelets lies above L, while at the main segment corresponding to v they lie below L

Color Twisting



the flip in the location of the cubelets makes it impossible to locally decide where the colored 3 cubelets should lie!

Claim1: This is W.L.O.G.

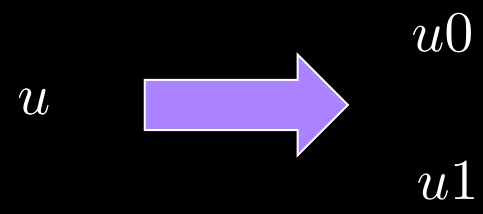
to resolve this we assume that all edges (u,v) of the PPAD graph join an odd u (as a binary number) with an even v (as a binary number) or vice versa

for even u's we place the pair of 3-colored cubelets below the main segment of u, while for odd u's we place it above the main segment

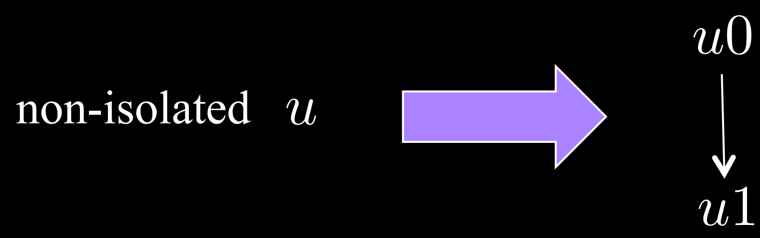
convention agrees with coloring around main segment of 0^n

Proof of Claim of Previous Slide

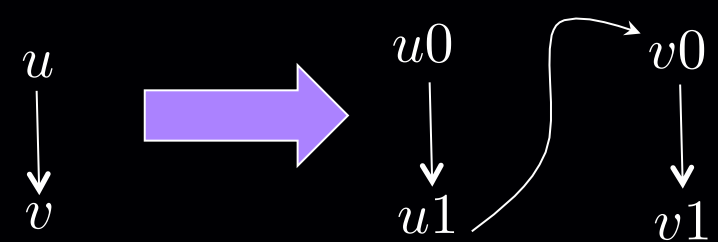
- Duplicate the vertices of the PPAD graph



- If node u is non-isolated include an edge from the 0 to the 1 copy



- Edges connect the 1-copy of a node to the 0-copy of its out-neighbor



Finishing the Reduction

A point in the cube is panchromatic iff it is the corner of some cubelet (i.e. it belongs to the subdivision of mutliples of 2^{-m}), and all colors are present in the cubelets containing this point.

Claim 1: *A point in the cube is panchromatic in the described coloring iff it is:*

- *an endpoint u_2' of a sink vertex u of the PPAD graph, or*
- *an endpoint u_1 of a source vertex $u \neq 0^n$ of the PPAD graph.*

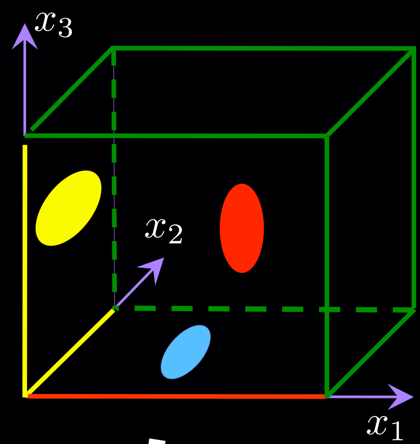
Claim 2: *Given the description P, N of the PPAD graph, there is a polynomial-size circuit computing the coloring of every cubelet* K_{ijk}.

PPAD

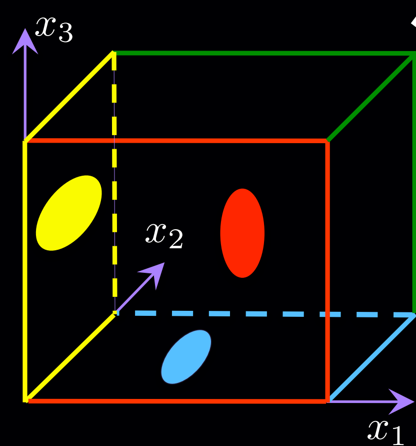
SPERNER

PPAD-completeness of BROUWER

(Special) SPERNER ⟹ *BROUWER*



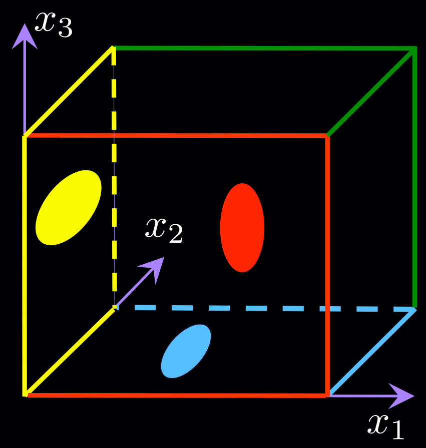
color fudging



Claim: Boundary coloring is not a legal Sperner coloring anymore, but no new panchromatic points were introduced by the modification.

Proof: The points that (were not but) could potentially become panchromatic after the modification are those with: x_1, x_2, or $x_3=1-2^{-m}$. But since the ambient space is colored green and the line L is far from the boundary, this won't happen.

(Special) SPERNER $\Rightarrow$ BROUWER



- Define BROUWER instance on the (slightly smaller) cube defined by the convex hull of the centers of the cubelets. This is thinner by 2^{-m} in each dimension.

- Convert color of K_{ijk} to direction of the displacement vector $f(x) - x$:

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

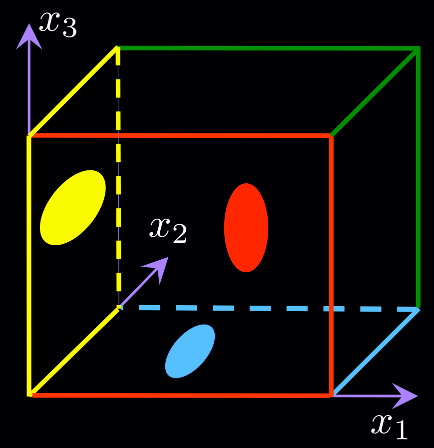
color 1 $\rightsquigarrow (1,0,0) \times \alpha$

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

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

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

(Special) SPERNER ⟶ BROUWER

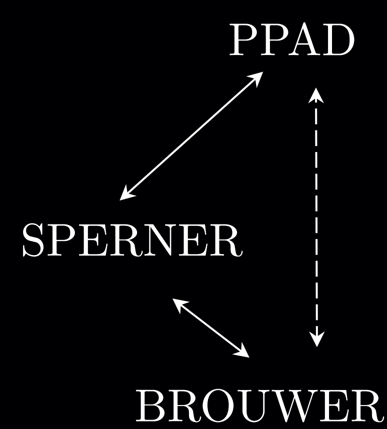


f is extended on the remaining cube by interpolation: The cube is triangulated in the canonical way. To compute the displacement of f at some point x, we find the simplex S to which x belongs. Then

if $x = \sum_{i=1}^{4} w_i \cdot x_i$, where x_i are the corners of S, we define :

$$f(x) - x := \sum_{i=1}^{4} w_i \cdot (f(x_i) - x_i)$$

Claim: Let x be a 2^{-3m}-approximate Brouwer Fixed Point of f. Then the corners of the simplex S containing x must have all colors.


PPAD
SPERNER
BROUWER

PPAD-completeness of NASH

(Special) BROUWER → NASH

Initial thoughts: *BROUWER, SPERNER as well as END OF THE LINE are defined in terms of explicit circuits (for computing the function value, coloring, or next/ previous nodes) specified in the description of the instance.*

In usual NP reductions, the computations performed by the gates in the circuits of the source problem need to somehow be simulated in the target problem.

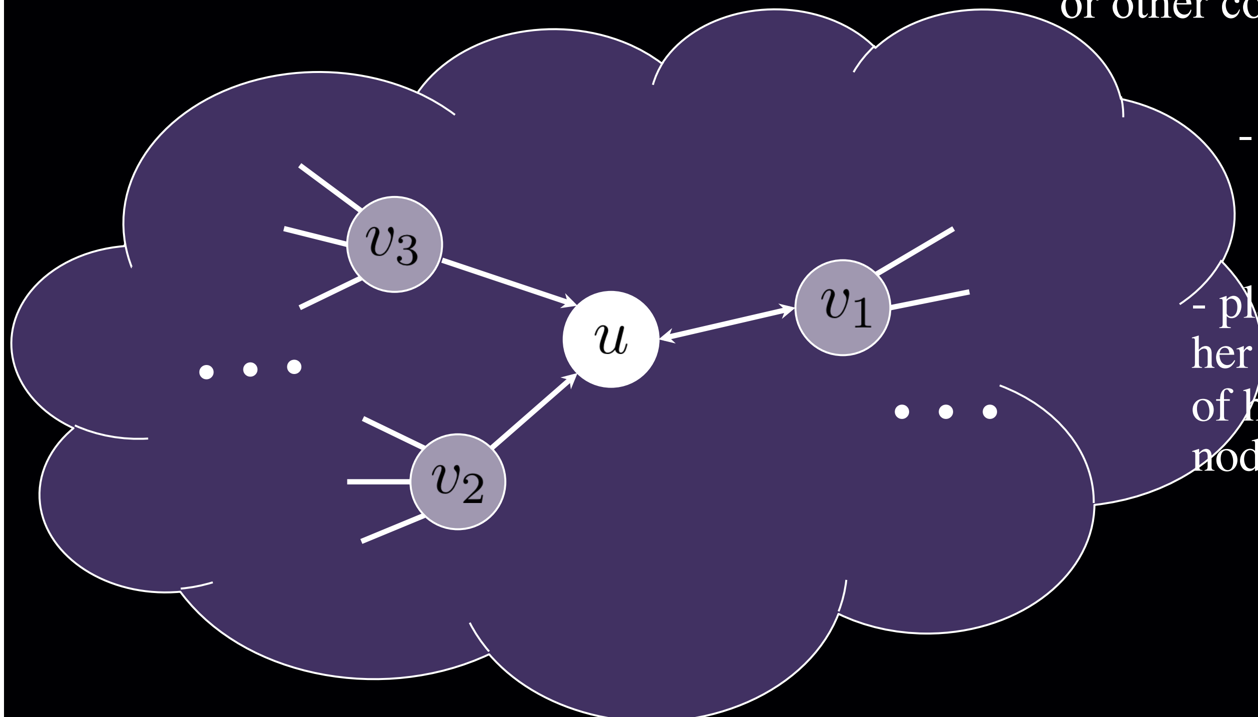
The trouble with NASH is that no circuit is explicitly given in the description of a game.

On the other hand, in many FNP-complete problems, e.g. Vertex Cover, we do not have a circuit in the definition of the instance (as is the case with Circuit Sat). But at least we have a combinatorial object to work with, such as a graph, which isn't the case here either…

(Special) BROUWER ➡ *NASH*

Introducing a graph structure, via *graphical games*.

defined to capture sparse player interactions, such as those arising under geographical, communication or other constraints.

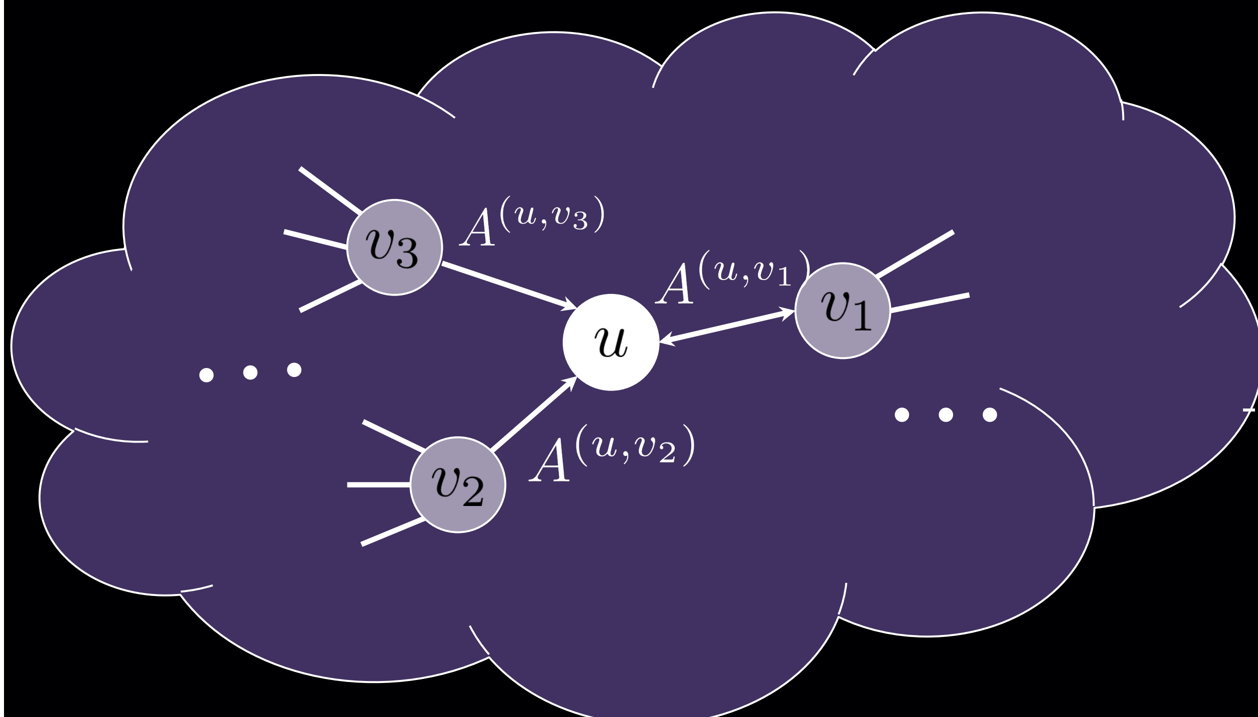


- players are nodes in a graph

- player's payoff is only affected by her own strategy and the strategies of her in-neighbors in the graph (i.e. nodes pointing to her)

(Special) BROUWER ➡ NASH

In fact, we restrict ourselves to a special class of graphical games, called graphical polymatrix games. These are graphical games with edge-wise separable utility functions.



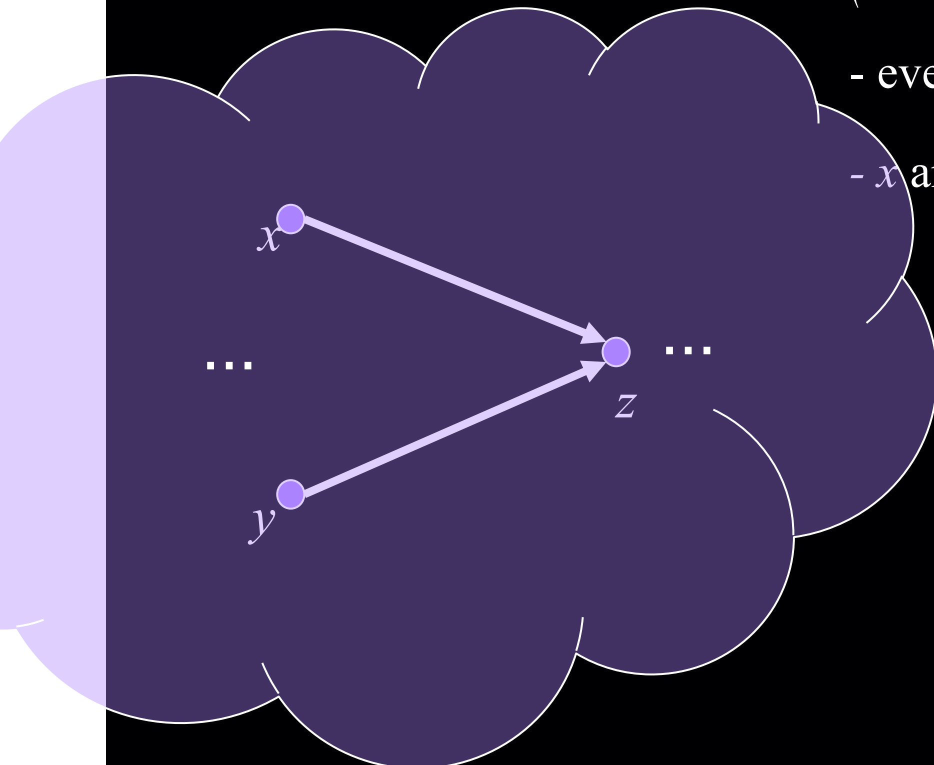
- edges are 2-player games

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

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

Can games perform conventional binary computation?

Can these games perform binary computation?



- 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, 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 where $\Pr[x{:}1], \Pr[y{:}1] \in \{0,1\}$, we have:

$$\Pr[z:1] = \Pr[x:1] \vee \Pr[y:1].$$

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

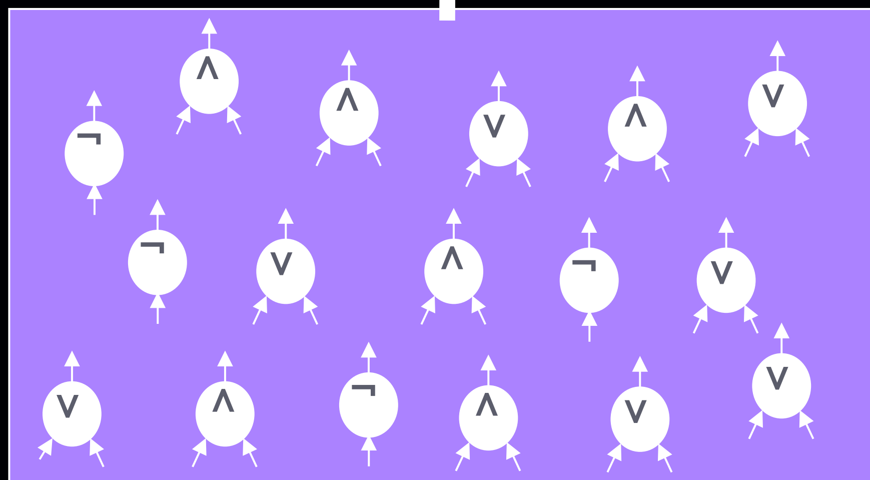
A possible PPAD-hardness reduction

does not exist unconditionally

if input is 0 enter a mode with no Nash eq.

"output" 1 if it is, and 0 if it is not

exists



check if the point $(i, j, k) \cdot 2^{-m}$ is panchromatic; all this is done in pure strategies, since the "input" to this part is in pure strategies

x_1 … x_m y_1 … y_m z_1 … z_m

interpret these pure strategies as the coordinates i, j, k of a point in the subdivision of the hypercube

exists

game gadget whose purpose is to have players $x_1, \ldots, z_m$ play **pure strategies** in any Nash equilibrium *bottom line:*

- *a reduction restricted to pure strategy equilibria is likely to fail*
- *real numbers seem to play a fundamental role in the reduction*

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

What in a Nash equilibrium is capable of storing reals?