Game Theory and Algorithms, Lake Como School of Advanced Studies
7-11 September 2015, Campione d'Italia

Computational complexity of solution concepts

UNIVERSITÀ DELLA CALABRIA
CAMPUS DI ARCAVACATA

Gianluigi Greco

Dept. of Mathematics and Computer Science
University of Calabria, Italy

Part I: P vs NP

One Basic Question…

- The lost letter…
  - Year: 1956
  - To: John von Neumann
- …and one question

P=NP?

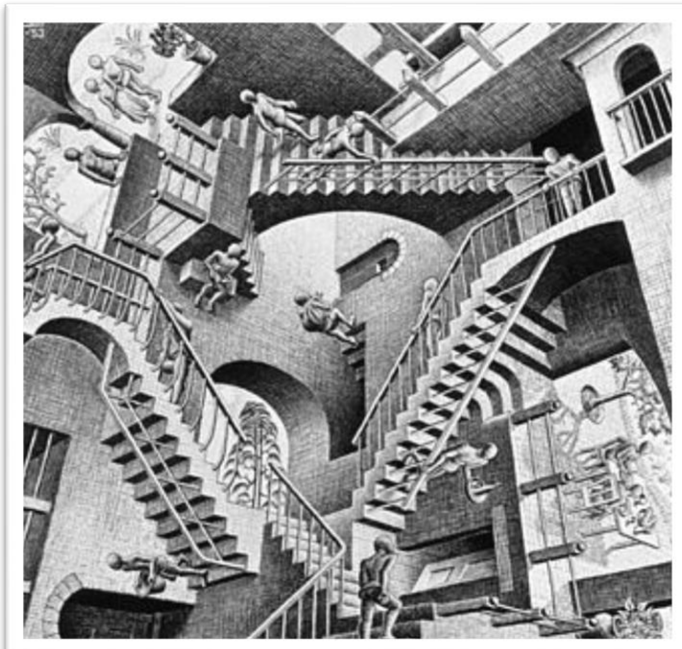
Kurt Gödel (1906-1978)

Millennium Prize Problems

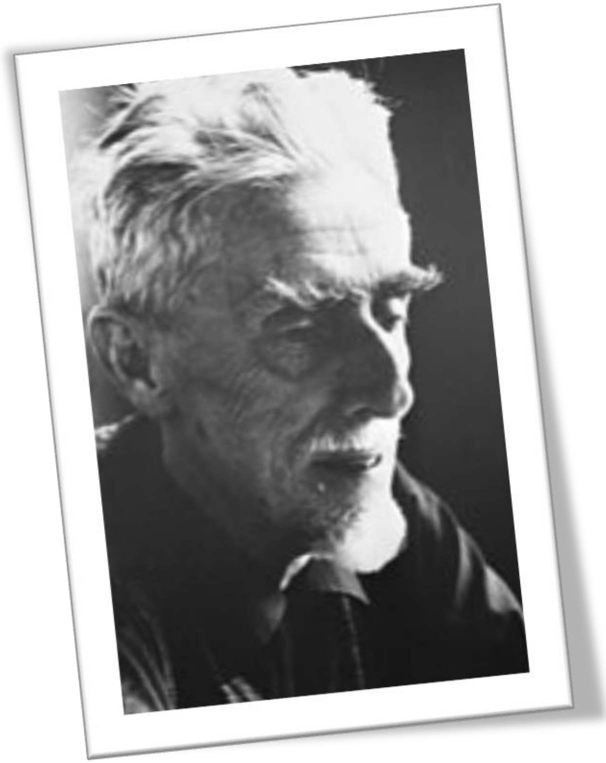
- Seven problems
  - **P versus NP**
  - Poincaré conjecture
  - Hodge conjecture
  - Riemann hypothesis
  - Yang–Mills existence and mass gap
  - Navier–Stokes existence and smoothness
  - Birch and Swinnerton-Dyer conjecture

CLAY MATHEMATICS INSTITUTE

About Clay



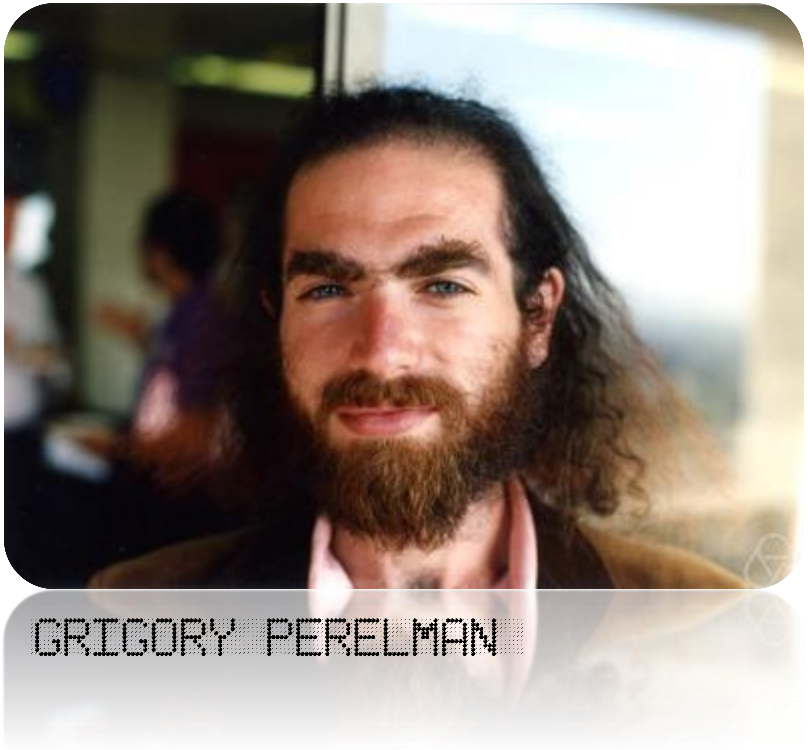
M. C. ESCHER
RELATIVITY, 1953



CLAY
MATHEMATICS
INSTITUTE

About the Problems

- One problem has been solved in 2006
  - Poincaré conjecture



- He did not want the prize…
- …and the Fields Medal

Problems

- Look for a document

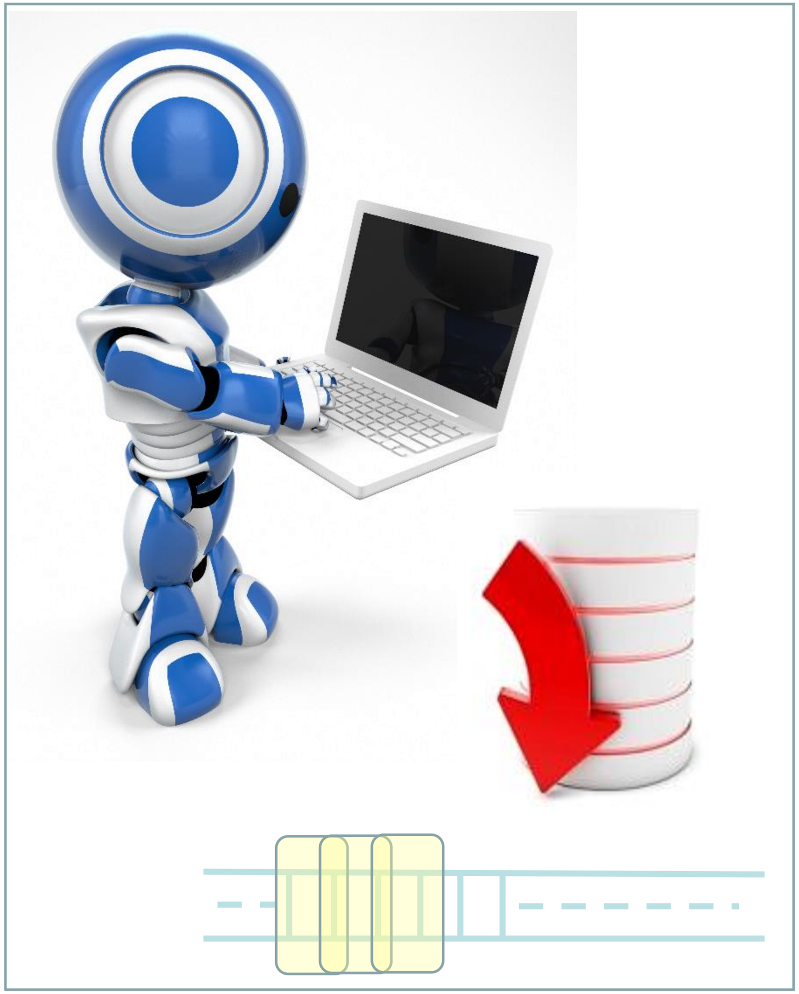
INPUT

OUTPUT

Algorithms

- Look for a document

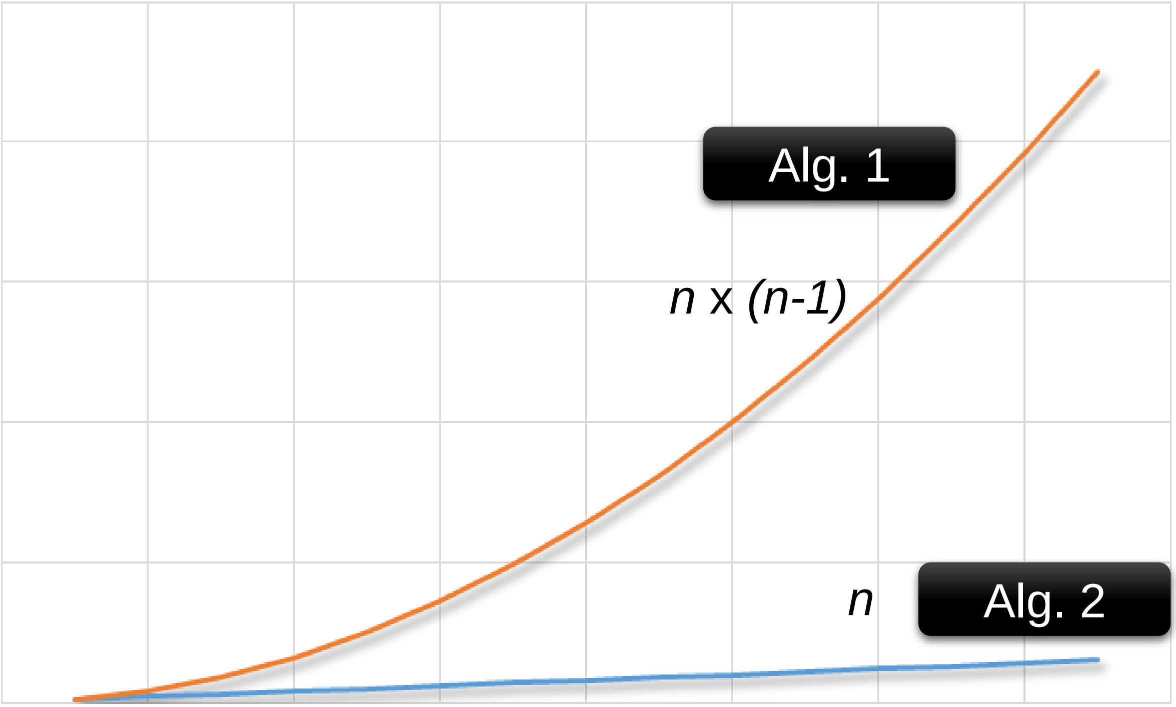
INPUT



OUTPUT

Comparison

- Algorithm 1
  - «quadratic»
- Algorithm 2
  - «linear»



The problem is in **P** (Polynomial)

Look at the Pictures

Look at the Pictures

Golden Ratio

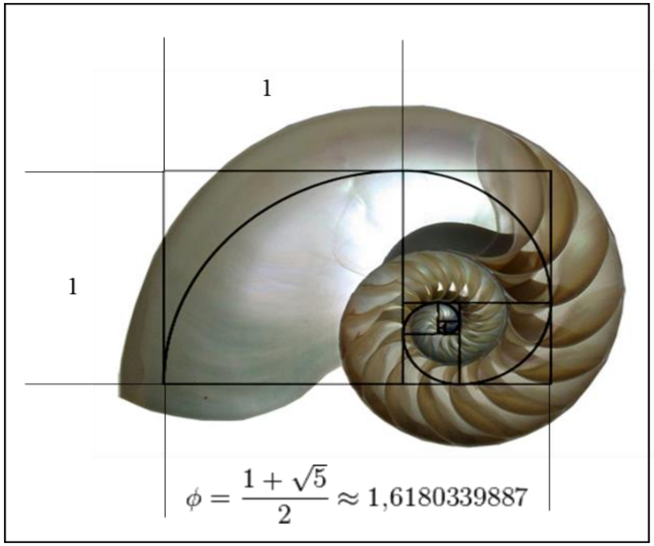
Golden Ratio

$$\frac{L}{R} = \frac{R+L}{L}$$

R=1

$$L = \frac{1+L}{L}$$

$$L^2-L-1=0$$



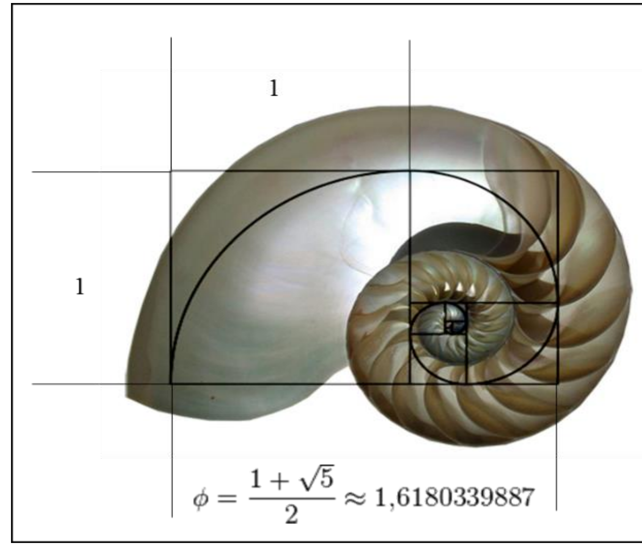
Golden Ratio

$$\frac{L}{R} = \frac{R+L}{L}$$

R=1

$$L = \frac{1+L}{L}$$

$$L^2\text{-}L\text{-}1=0$$



- Does there exist a solution?
- More generally, take L_1, L_2,…L_n and answer questions such as:
  - Is ithere some value for L_1 such that for all values for L_2…

Golden Ratio

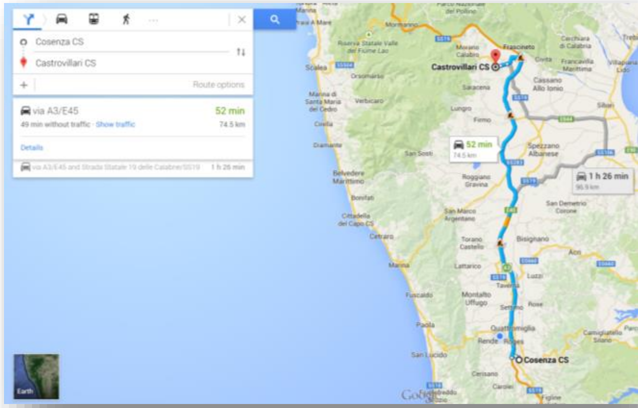
- Provably **EXP**onential
- Every algorithm takes $\mathbf{10^n}$ operations in the worst case

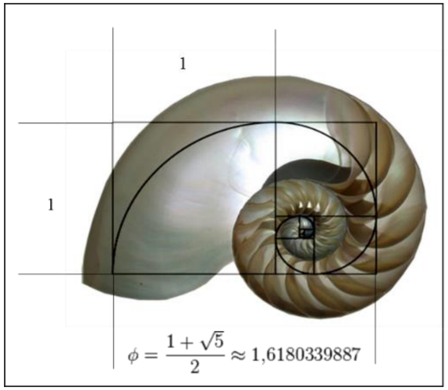
- Does there exist a solution?
- More generally, take L_1, L_2,…L_n and answer questions such as:
  - Is ithere some value for L_1 such that for all values for L_2…

Orders of Magnitude

- 10^4: There are 20,000–40,000 distinct Chinese characters
- 10^5 : 67,000 words in James Joyce's Ulysses
- 10^6 : As of August 31, 2015, Wikipedia contains approximately 4956000 articles in the English language
- 10^9: Approximate population of India in 2011
- 10^{14}: Cells in the human body
- 10^{21}: Estimated number of observable stars
- 10^{80}: Atoms in the Universe

Classes of Problems

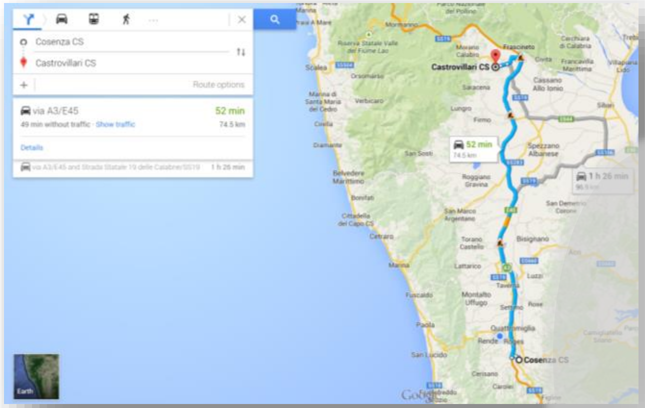


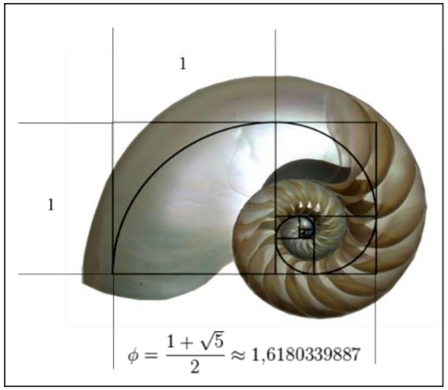


P

EXP

Classes of Problems





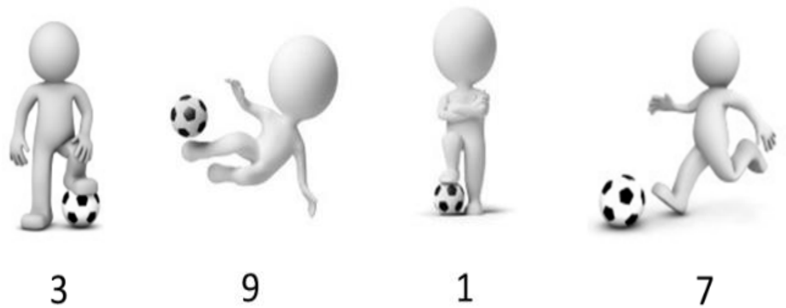
P

EXP

NP

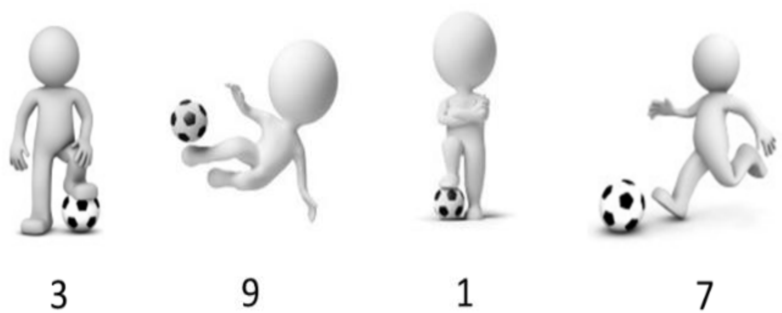
A Simple Problem for NP

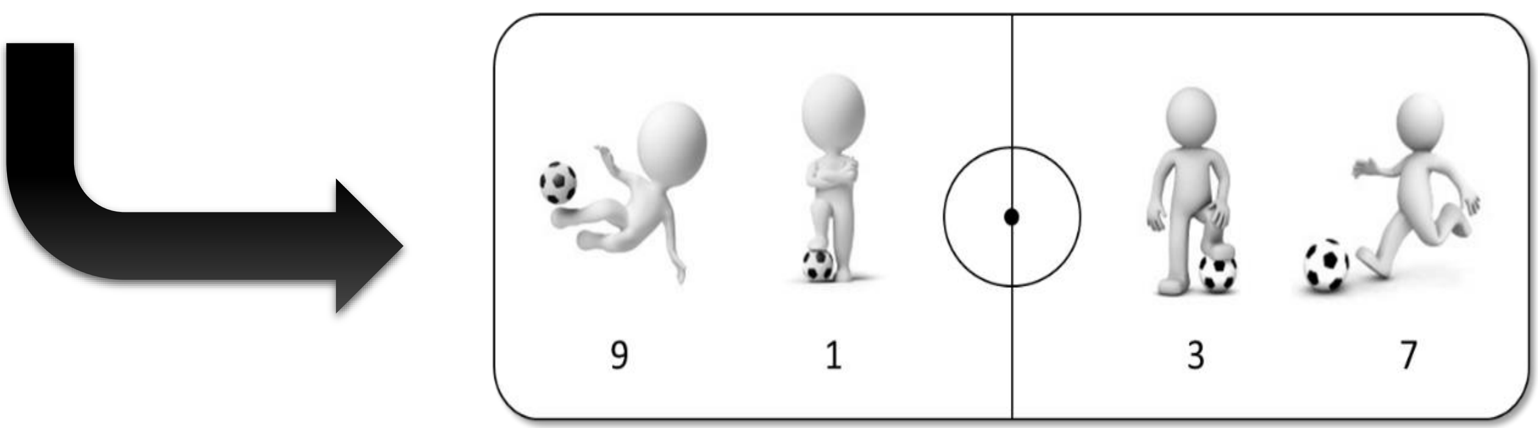
- We would like to set up two teams
- Goal
  - The teams should be «balanced»



A Simple Problem for NP

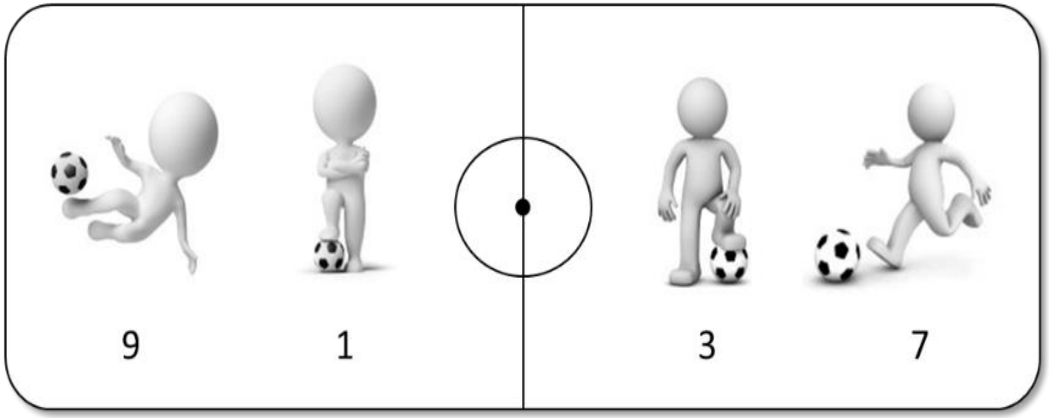
- We would like to set up two teams
- Goal
  - The teams should be «balanced»





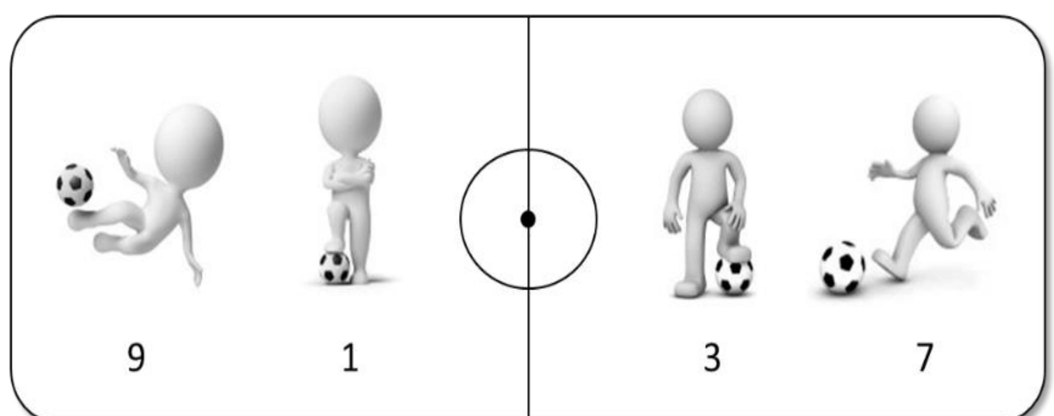
A Simple Problem for NP

- We would like to set up two teams
- Goal
  - The teams should be «balanced»
- Algorithm
  - Consider all possible teams



A Simple Problem for NP

- We would like to set up two teams
- Goal
  - The teams should be «balanced»
- Algorithm
  - Consider all possible teams

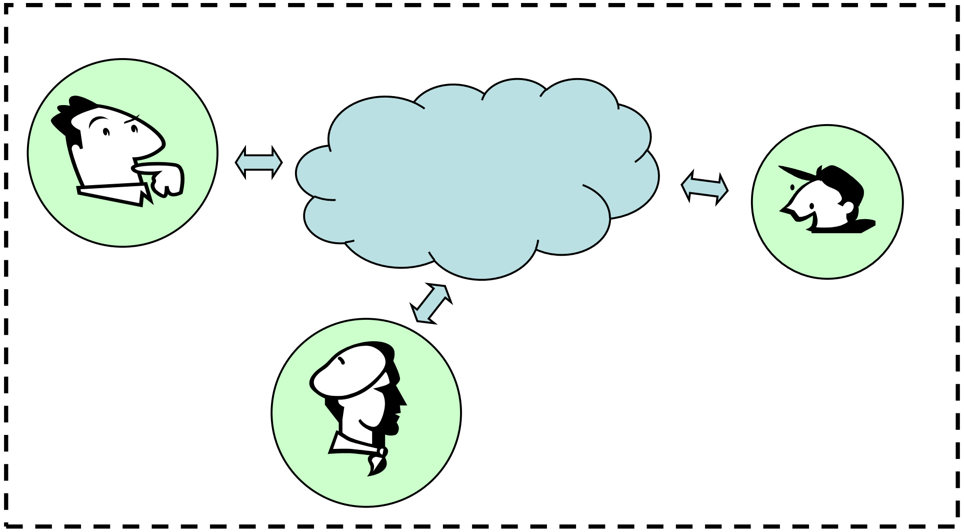


Further Examples

- Scheduling
- Planning
- Logistics
- Crypto
- …Game Theory

Part II: Nash Equilibria

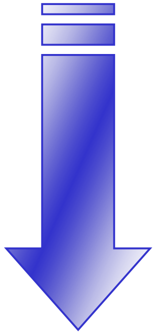
Game Theory (in a Nutshell)



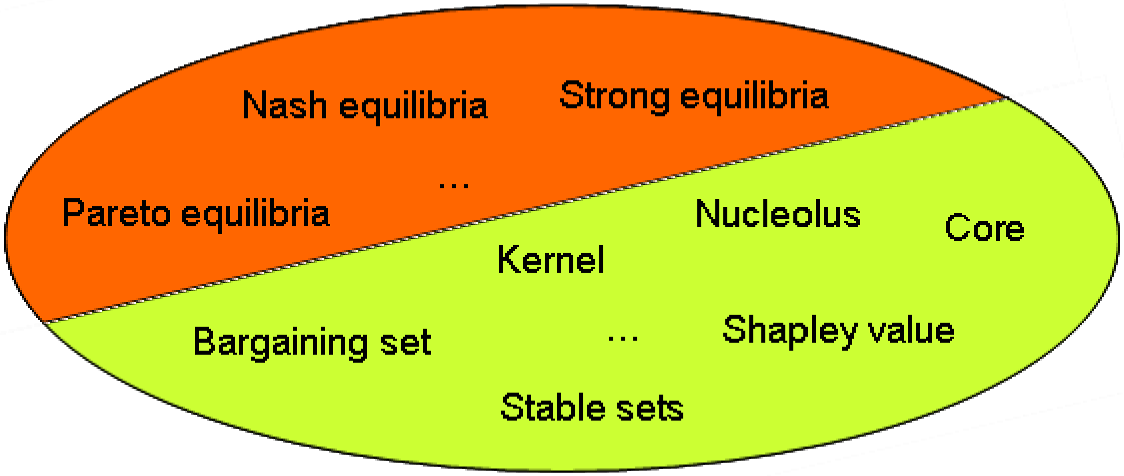
Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

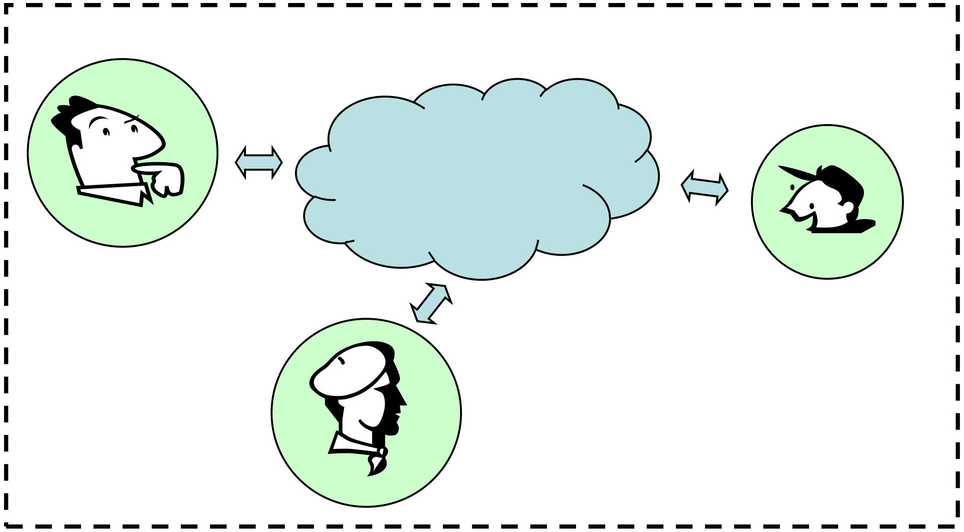
Which actions have to be performed?



Solution Concepts



Game Theory (in a Nutshell)

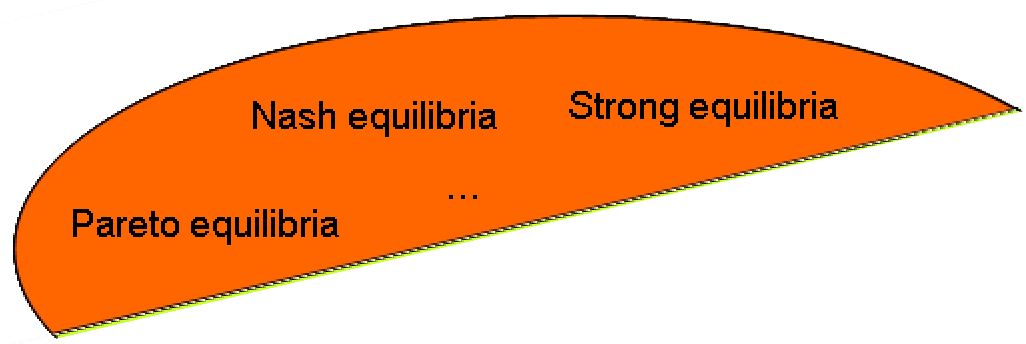


Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

Which actions have to be performed?

Solution Concepts



Non-Cooperative Games$_{(1/3)}$

Payoff maximization problem

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

Bob	John goes *out*	John stays at *home*
out	2	0
home	0	1

John	Bob goes *out*	Bob stays at *home*
out	1	1
home	0	0

Non-Cooperative Games$_{(2/3)}$

Payoff maximization problem

Nash equilibria

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

Bob	John goes *out*	John stays at *home*
out	2	0
home	0	1

John	Bob goes *out*	Bob stays at *home*
out	1	1
home	0	0

Non-Cooperative Games$_{(2/3)}$

Payoff maximization problem

Nash equilibria

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

Bob	John goes *out*	John stays at *home*
out	2	0
home	0	1

John	Bob goes *out*	Bob stays at *home*
out	1	1
home	0	0

Non-Cooperative Games$_{(2/3)}$

Payoff maximization problem

Nash equilibria

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

Bob	John goes *out*	John stays at *home*
out	2	0
home	0	1

John	Bob goes *out*	Bob stays at *home*
out	1	1
home	0	0

Non-Cooperative Games$_{(2/3)}$

Payoff maximization problem

Nash equilibria

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

Bob	John goes *out*	John stays at *home*
out	**2**	0
home	0	1

John	Bob goes *out*	Bob stays at *home*
out	**1**	1
home	0	0

Non-Cooperative Games$_{(3/3)}$

Payoff maximization problem

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

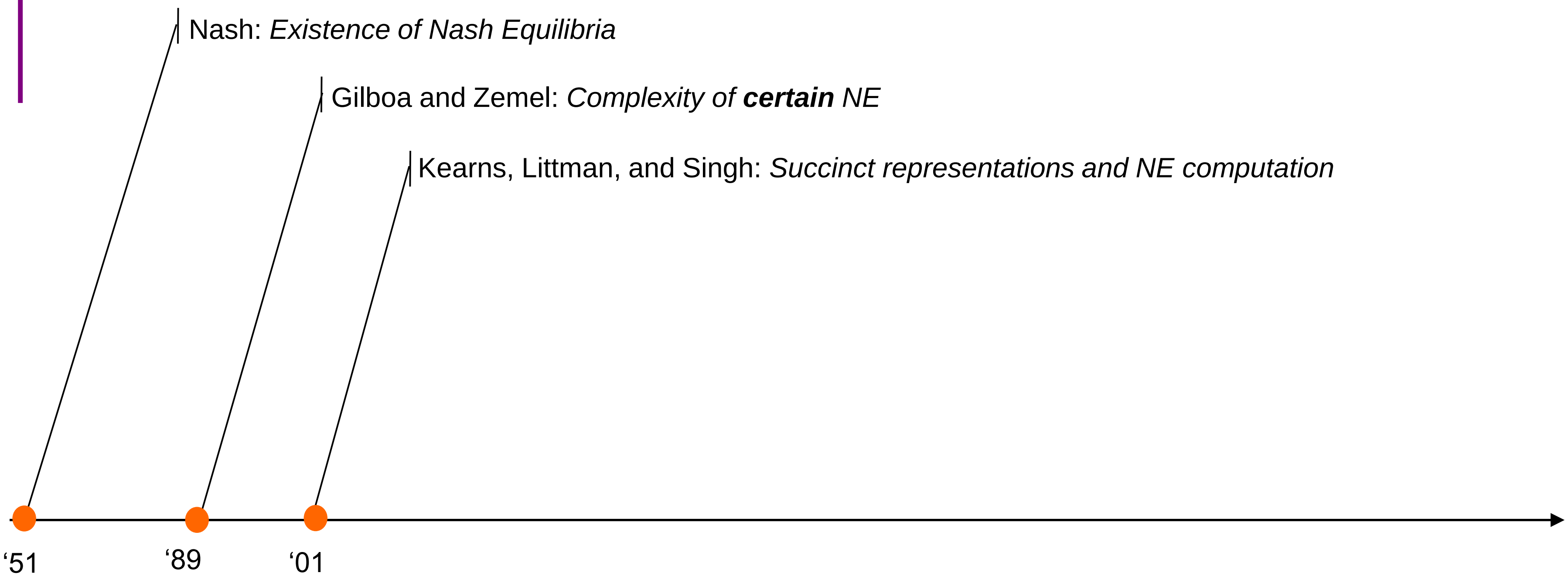
Nash equilibria

---

pure **Nash equilibria**

Every game admits a *mixed* Nash equilibrium,

- **where players chose their strategies according to probability distributions**



Nash: *Existence of Nash Equilibria*
Gilboa and Zemel: *Complexity of **certain** NE*
Kearns, Littman, and Singh: *Succinct representations and NE computation*
'51
'89
'01

Succint Game Representations

- Players:
  - Maria, Francesco
- Choices:
  - movie, opera

If 2 players, then size = 2^2

Maria	Francesco, *movie*	Francesco, *opera*
movie	2	0
opera	0	1

Succint Game Representations

- Players:
  - Maria, Francesco, Paola
- Choices:
  - movie, opera

If 2 players, then size = 2^2

If 3 players, then size = 2^3

Maria	F_{movie} *and* P_{movie}	F_{movie} *and* P_{opera}	F_{opera} *and* P_{movie}	F_{opera} *and* P_{opera}
movie	2	0	2	1
opera	0	1	2	2

Succint Game Representations

- Players:
  - Maria, Francesco, Paola, Roberto, and Giorgio
- Choices:
  - movie, opera

If 2 players, then size = 2^2

If 3 players, then size = 2^3

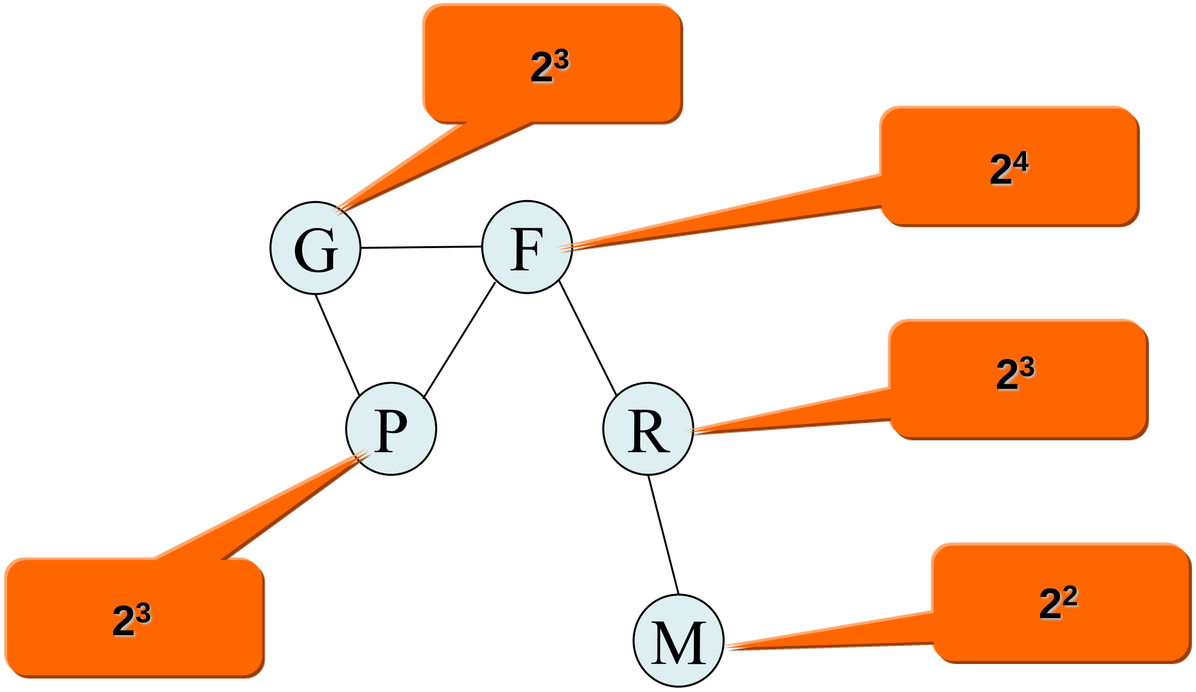
…

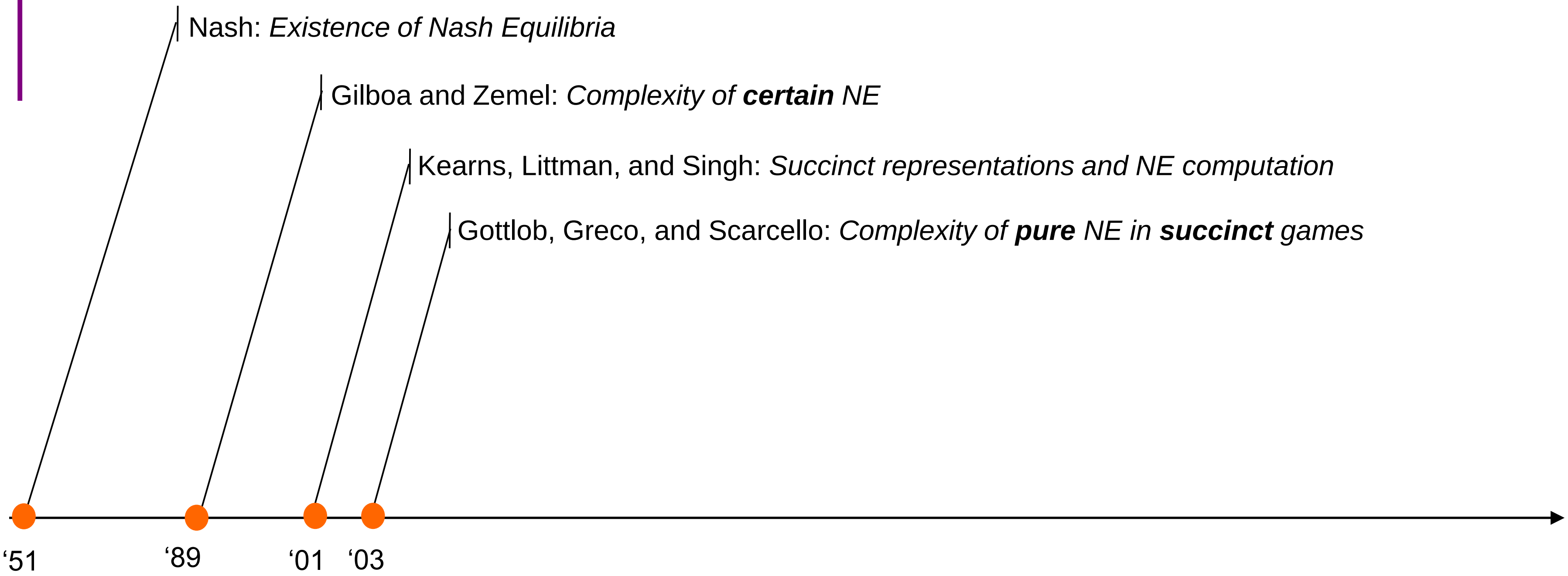
If N players, then size = 2^N

Maria	F_{movie} *and* P_{movie} *and* R_{movie} *and* G_{movie}	…………………………		
movie	2	……..	……..	……..
opera	0	……..	……..	……..

Succinct Game Representations

- Players:
  - Francesco, Paola, Roberto, Giorgio, and Maria
- Choices:
  - movie, opera




Nash: *Existence of Nash Equilibria*
Gilboa and Zemel: *Complexity of **certain** NE*
Kearns, Littman, and Singh: *Succinct representations and NE computation*
Gottlob, Greco, and Scarcello: *Complexity of **pure** NE in **succinct** games*
‘51
‘89
‘01
‘03

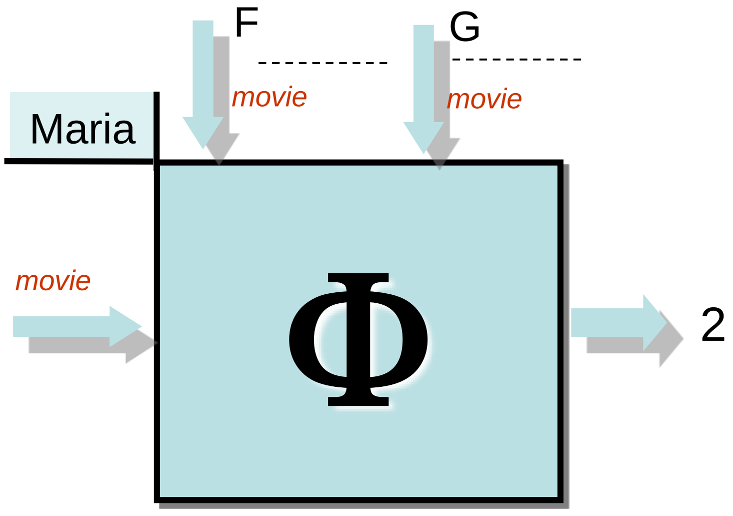
Complexity of Pure Nash Equilibria$_{(1/3)}$

- Game Representation
  - Tables
  - Arbitrary Functions

Maria	F_{movie} *and* P_{movie}	*and* R_{movie} *and* G_{movie}	…………………………	
movie	2	……..	……..	……..
opera	0	……..	……..	……..

Complexity of Pure Nash Equilibria$_{(1/3)}$

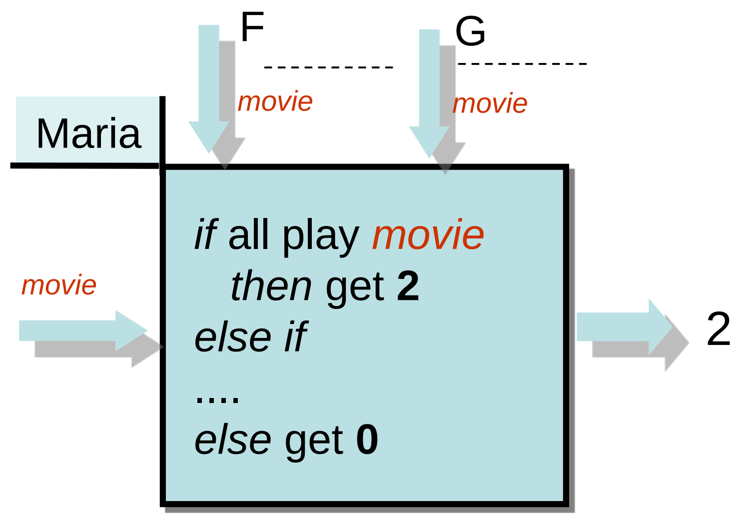
- Game Representation
  - Tables
  - Arbitrary Functions



Maria	F$_{movie}$ *and* P$_{movie}$ *and* R$_{movie}$ *and* G$_{movie}$	…………………………..		
movie	2	……..	……..	……..
opera	0	……..	……..	……..

Complexity of Pure Nash Equilibria$_{(1/3)}$

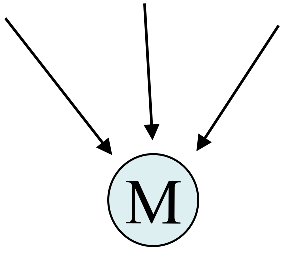
- Game Representation
  - Tables
  - Arbitrary Functions



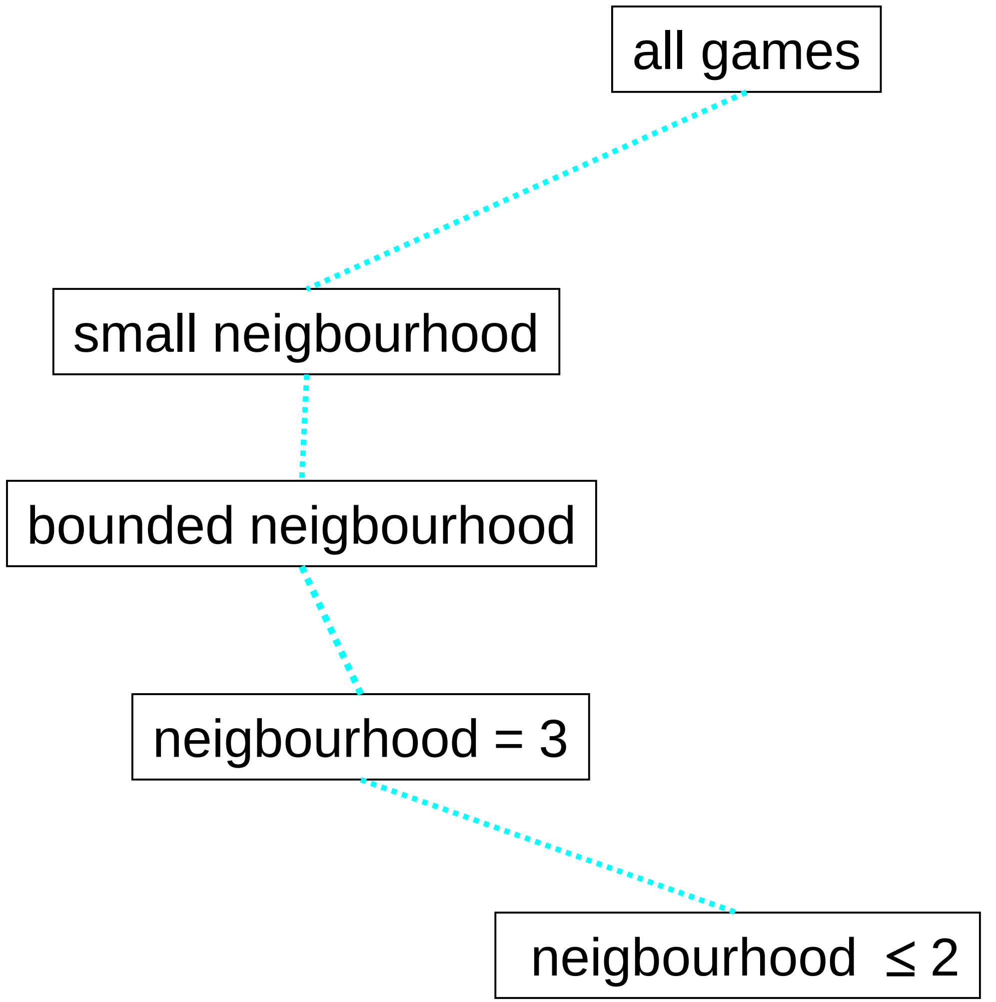
Maria	F_{movie} *and* P_{movie} *and* R_{movie} *and* G_{movie}	………………………...		
movie	2	……..	……..	……..
opera	0	……..	……..	……..

Complexity of Pure Nash Equilibria$_{(1/3)}$

- Game Representation
  - Tables
  - Arbitrary Functions
- Neighborood
  - Arbitrary
  - Small (i.e., log)
  - Bounded (i.e., constant)



Complexity of Pure Nash Equilibria$_{(3/3)}$



Complexity of Pure Nash Equilibria$_{(3/3)}$

THE BAD NEWS:

NP-c all games

NP-c small neigbourhood

NP-c bounded neigbourhood

NP-c neigbourhood = 3

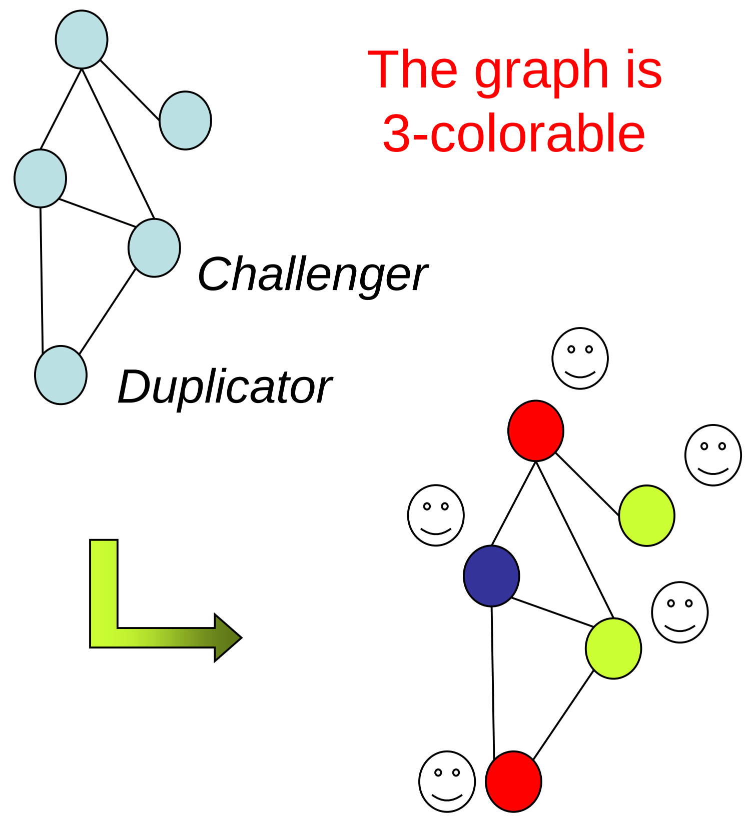
polynomial neigbourhood ≤ 2

Hard Games: NP-hardness

Theorem Deciding whether a game has pure Nash equilibrium is NP-complete. Hardness holds even if the game is in GNF, and if it has 3-bounded neighborhood.

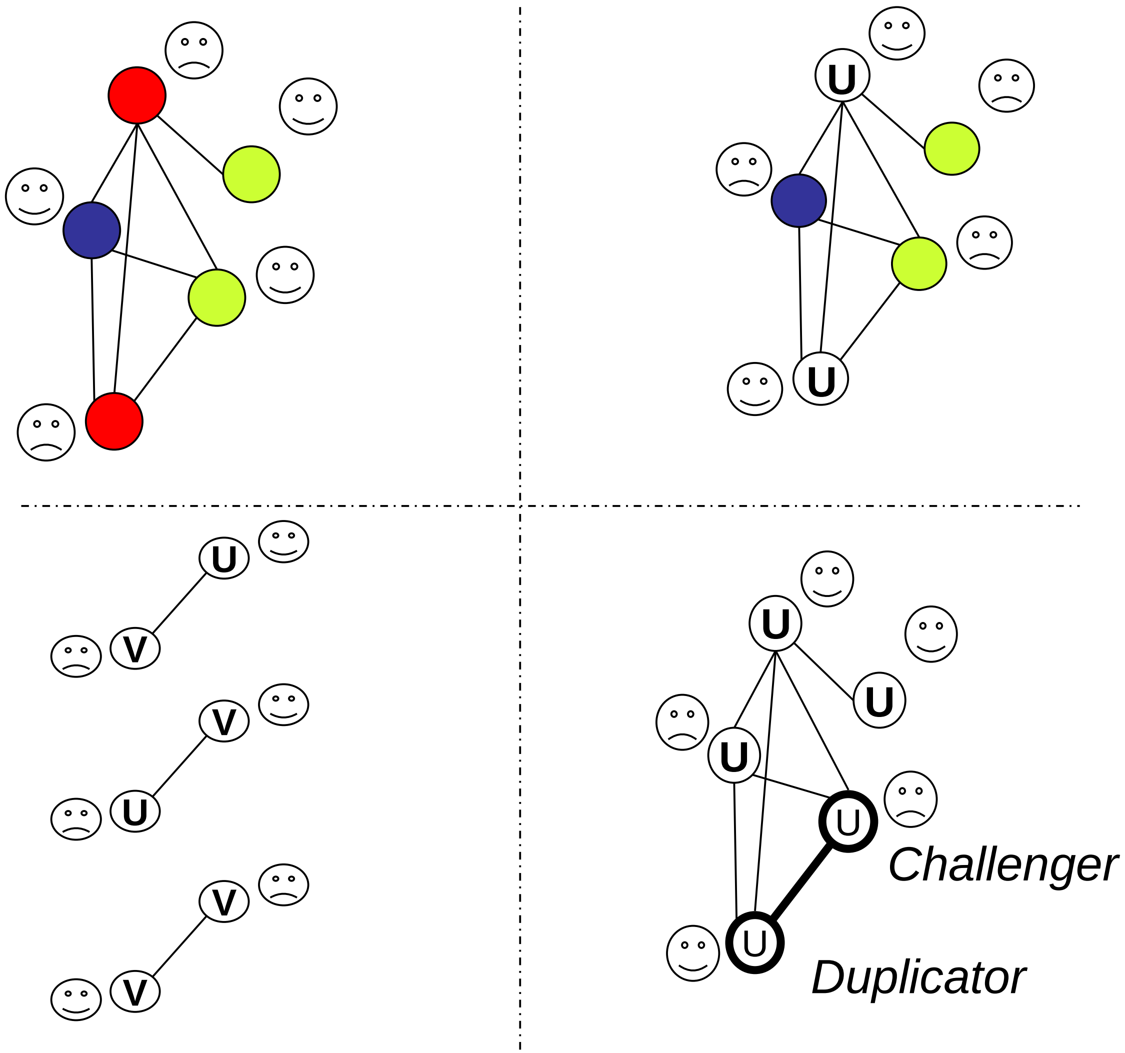
Reduction from 3-colorability

- **Players**: Nodes
- **Actions**: Colors + {U,V}
- **Utility Functions**: the players have an incentive to play a color different from the ones played by the neighbors
- ***Challenger*** and ***Duplicator*** are distinguished (connected) players such that:
  - *C* wants to play an action different from *D*;
  - *D* want to play either a color different from *C*, or the same action in {U,V}.



Hard Games: NP-hardness

The graph is **not** 3-colorable:

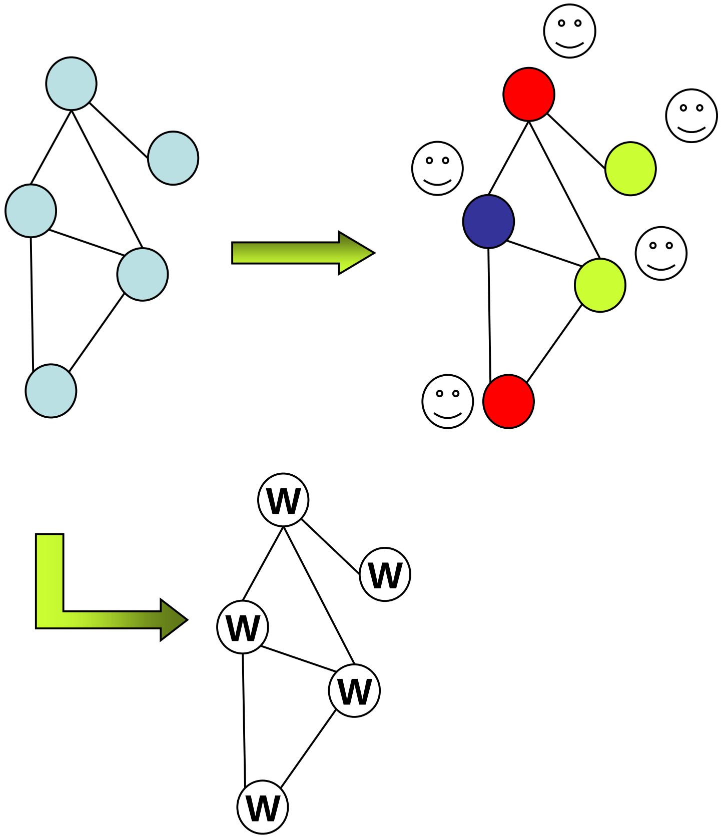


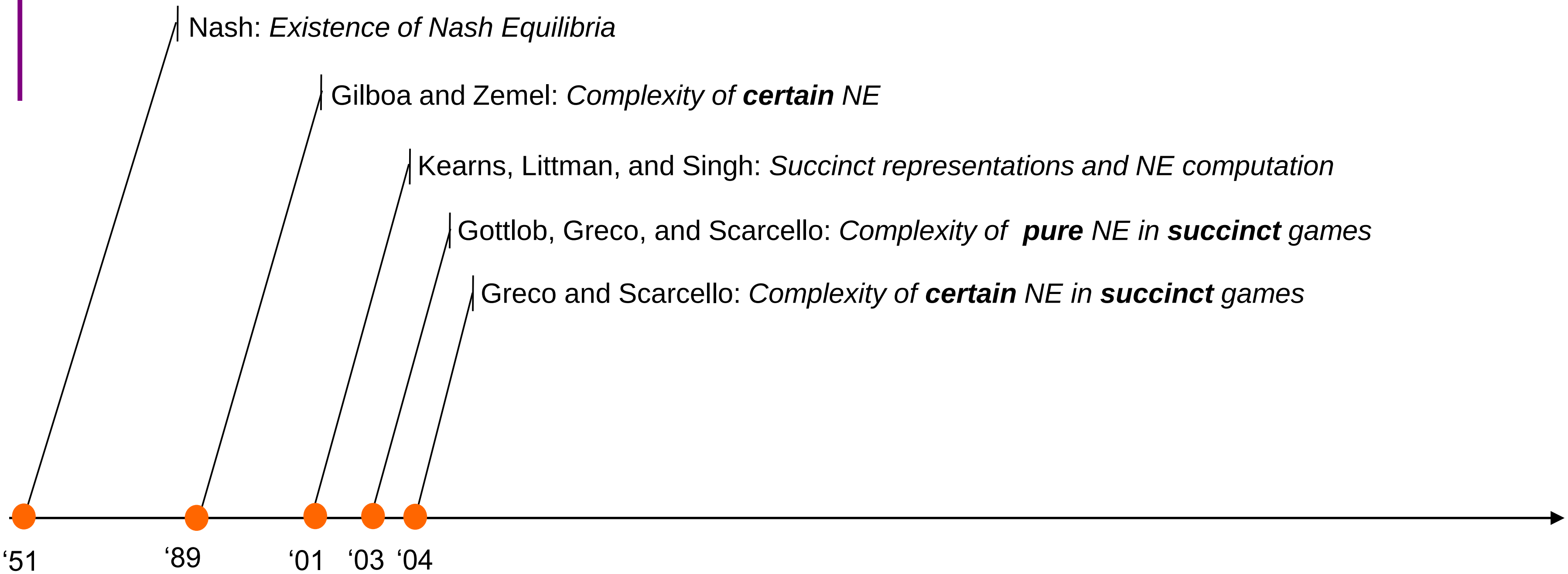
- Adjacent players playing the same color have an incentive to play U.
- The neighbors of players playing U have an incentive to play U, in their turn.
- Challenger want to plays an action different of Duplicator.
- No Nash equilibria exist.

Hard Games: coNP-hardness

Theorem Deciding whether a global strategy **x** is a Pareto (Strong) Nash equilibrium is coNP-complete. Hardness holds even if **x** is a Nash equilibrium, the game is in GNF, and if it has 3-bounded neighborhood.

- **Reduction from 3-non-colorability**
- **The same construction as above except:**
  - Each player may play also W, and has an incentive in such choice if all her neighbors play W, too.
- **There is alway a Nash equilibrium where all players play W**
  - Utility functions are such that it this equilibrium is not preferred to the equilibrium (if any) corresponding to a 3-coloring.
  - This equilibrium is Pareto iff the graph is not 3-colorable.





Nash: *Existence of Nash Equilibria*
Gilboa and Zemel: *Complexity of **certain** NE*
Kearns, Littman, and Singh: *Succinct representations and NE computation*
Gottlob, Greco, and Scarcello: *Complexity of **pure** NE in **succinct** games*
Greco and Scarcello: *Complexity of **certain** NE in **succinct** games*
'51
'89
'01
'03
'04

Constrained Nash Equilibria$_{(1/3)}$

Bob	John goes *out*	John stays at *home*
out	1	0
home	0	1

John	Bob goes *out*	Bob stays at *home*
out	0	1
home	1	0

Constrained Nash Equilibria $_{(1/3)}$

- Computing "any" Nash equilibria might not be enough
  - E.g., multi-agent planning, routing protocols, etc.
- What if we ask for "certain types of equilibrium"?
  - *Bob gets at least 1*
  - *The best social welfare*
  - *Maria cannot go to the opera*
  - *…*

Bob	John goes *out*	John stays at *home*
out	1	0
home	0	1

John	Bob goes *out*	Bob stays at *home*
out	0	1
home	1	0

Constrained Nash Equilibria$_{(2/3)}$

Evaluation functions

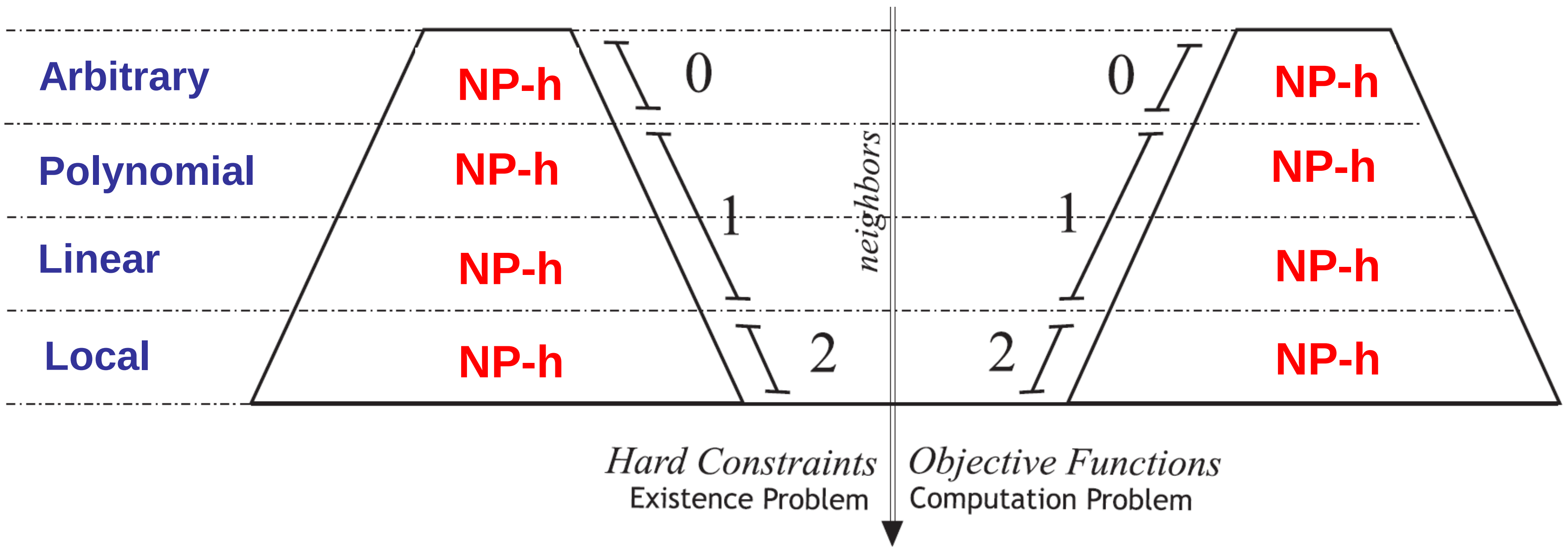
- **F_P:** polynomial-time computable functions, associating real numbers with each combined strategy of players in **P** and their neighbors

- Examples
  - *Let* $A_{\{G,P\}}$ return the minimum payoff between Giorgio and Paola
    - $A_{\{G,P\}} > 1$ is a guarantee for *G* and *P*
  - *Let* $B_{\{F,P,R,G,M\}}$ return the sum of the payoffs of all players
    - By maximizing $B_{\{F,P,R,G,M\}}$, we optimize the social welfare

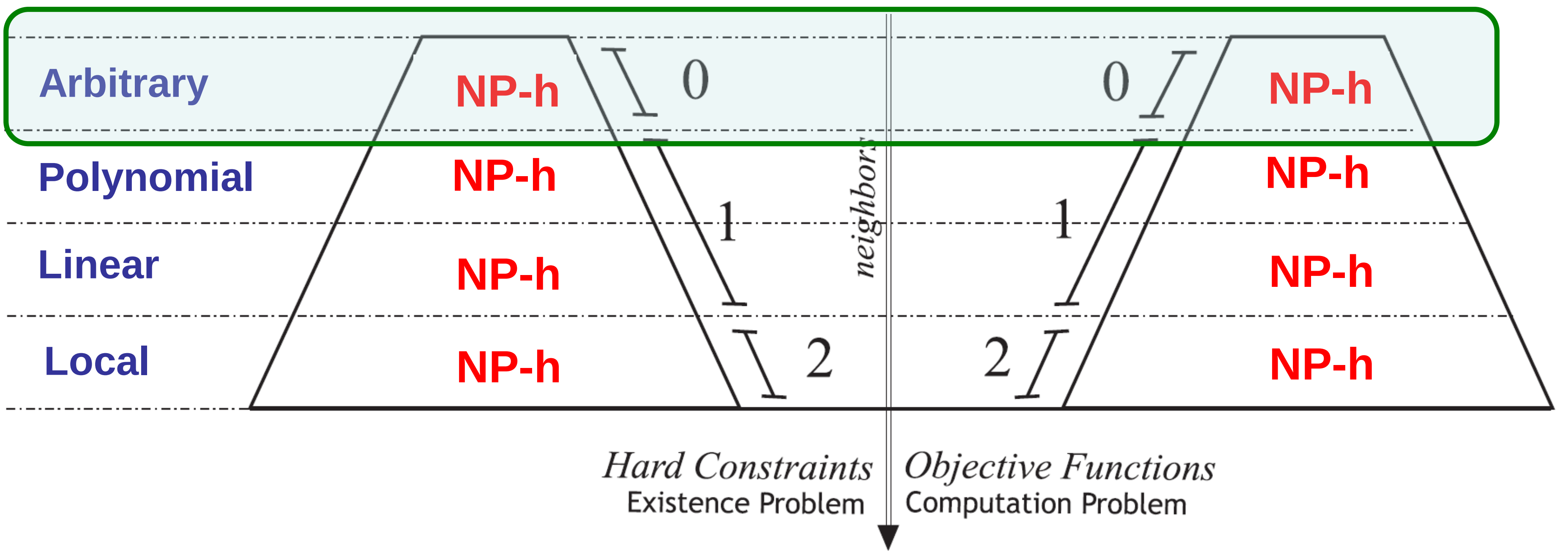
Hard Constraints

Objective Functions

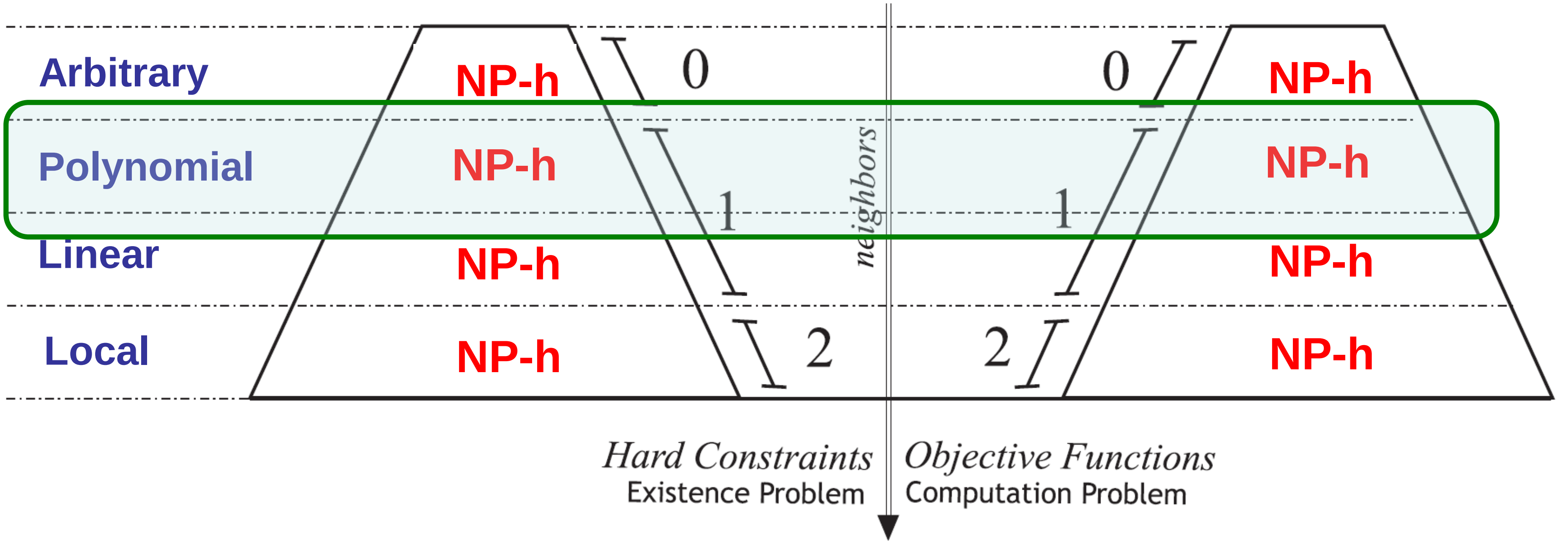
Constrained Nash Equilibria$_{(3/3)}$



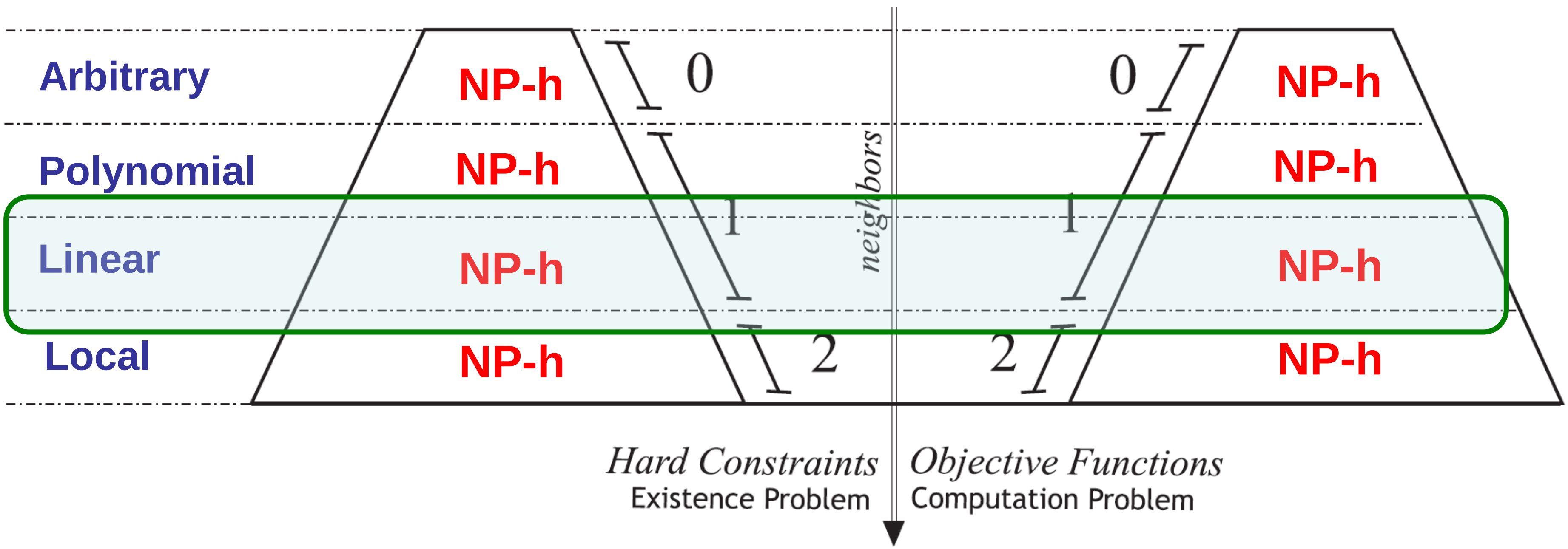
Constrained Nash Equilibria$_{(3/3)}$

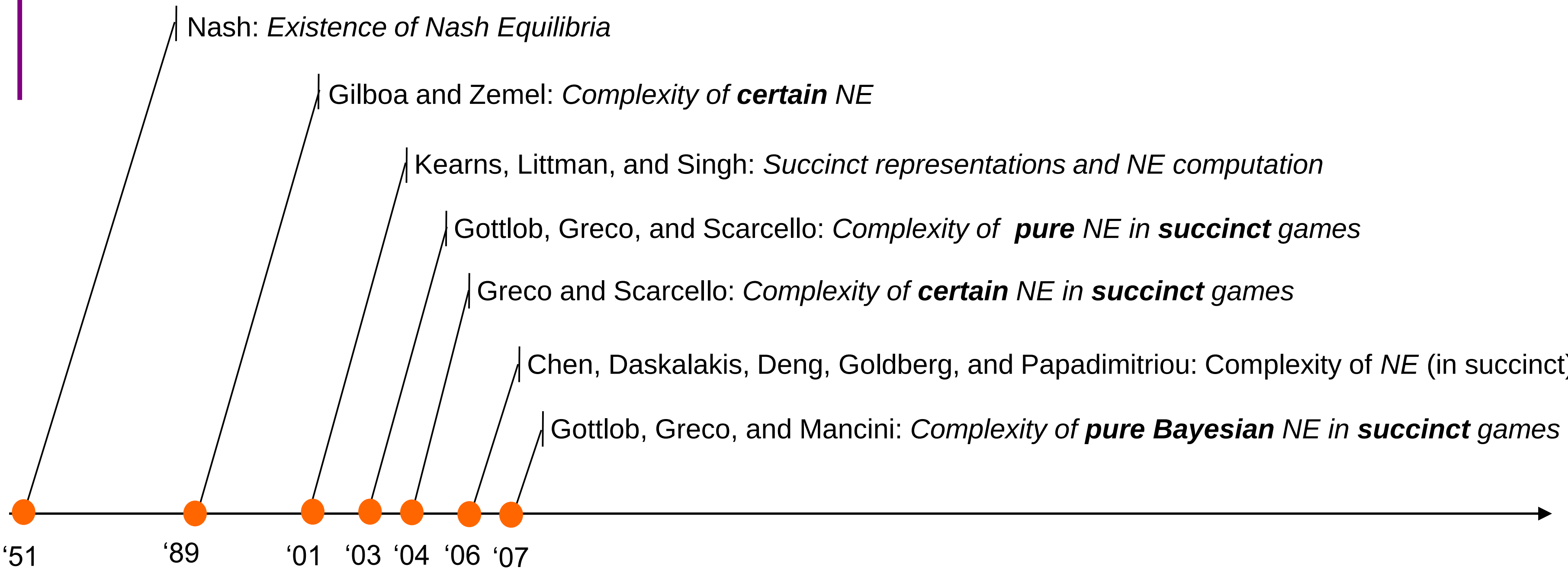


Constrained Nash Equilibria(3/3)



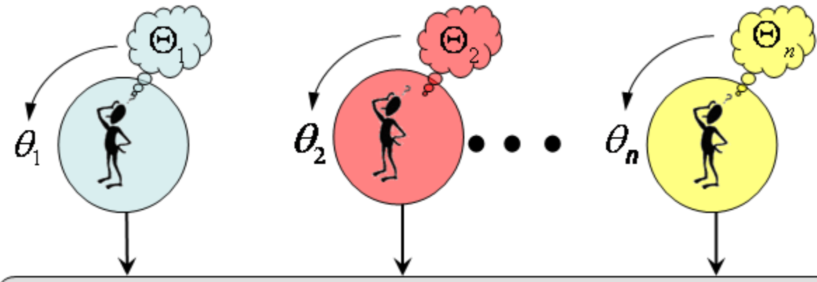
Constrained Nash Equilibria$_{(3/3)}$





Nash: Existence of Nash Equilibria
Gilboa and Zemel: Complexity of certain NE
Kearns, Littman, and Singh: Succinct representations and NE computation
Gottlob, Greco, and Scarcello: Complexity of pure NE in succinct games
Greco and Scarcello: Complexity of certain NE in succinct games
Chen, Daskalakis, Deng, Goldberg, and Papadimitriou: Complexity of NE (in succinct)
Gottlob, Greco, and Mancini: Complexity of pure Bayesian NE in succinct games
‘51
‘89
‘01
‘03
‘04
‘06
‘07

Bayesian Nash Equilibria$_{(1/3)}$



Bayesian Games (Harsany [1968])

- The **Nature** reveals a type
- Each player does not know the type of the other players, but... a probability distribution

Bob	John goes *out*	John stays at *home*
out	2	0
home	0	1

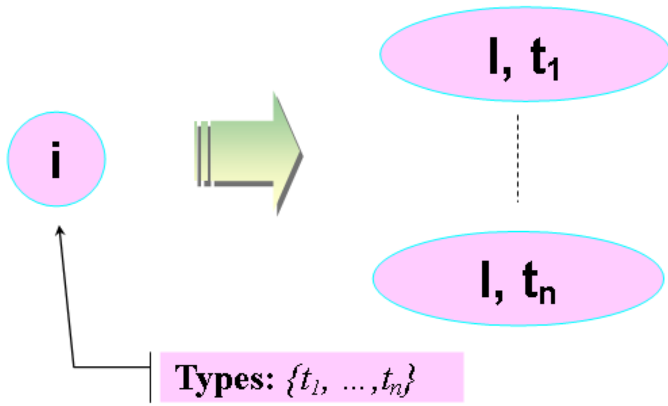
Type 1

Bob	John goes *out*	John stays at *home*
out	0	1
home	1	0

Type 2

Bayesian Nash Equilibria$_{(2/3)}$

Type-agent representation



$$v_{(i,t_i)}(\boldsymbol{a}) = \sum_{\boldsymbol{t}^{-i} \in \bigtimes_{j \in N-\{i\}} T_j} p_i(\boldsymbol{t}^{-i} | t_i)\; u_i\left(\boldsymbol{a}', \langle \boldsymbol{t}^{-i}; t_i \rangle\right)$$

BEs of the original game are in one-to-one correspondence with Nash Equilibria of the new game with complete information.

- The transformation:
  - Is feasible in polynomial time
  - Preserves the neighboord
  - Preserves the structural properties

Easy cases are preserved

Bayesian Nash Equilibria$_{(3/3)}$

PBE is **NP**-*complete, if the number of players is constant.*

PBE is **PP**-*hard, Hardness holds even for games where players have just two types.*

PBE is **PP**-*complete, in the fixed precision setting.*

PP: languages that can be decided by a nondeterministic Turing machine in polynomial time, where the acceptance condition is that a majority (more than half) of computation paths accept.

$\Rightarrow$ NP $\subseteq$ **PP** $\subseteq$ PSPACE

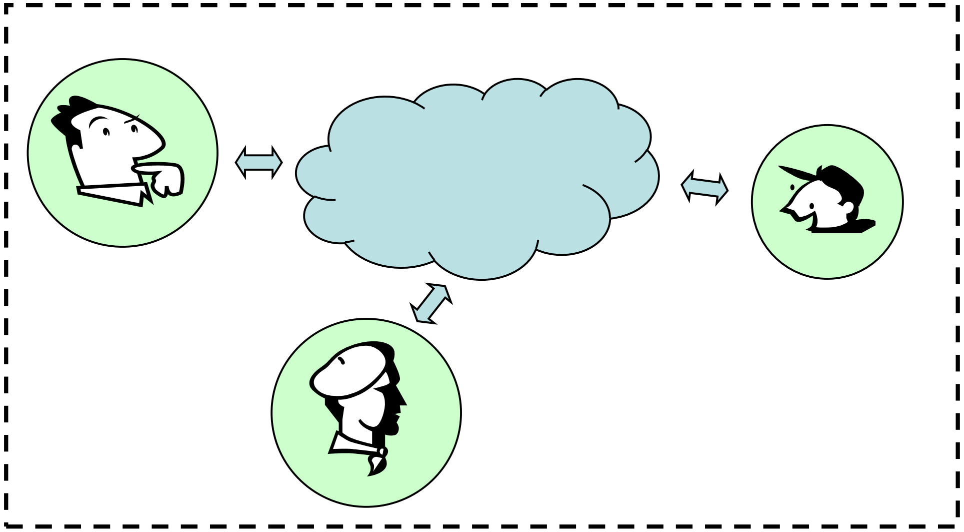
Canonical Problem: MAJ-SAT

Input: a formula Φ

Output: *true* $\Leftrightarrow$ Φ is satisfied by more than half of possible assignments

Part III: Nucleolus

Game Theory (in a Nutshell)

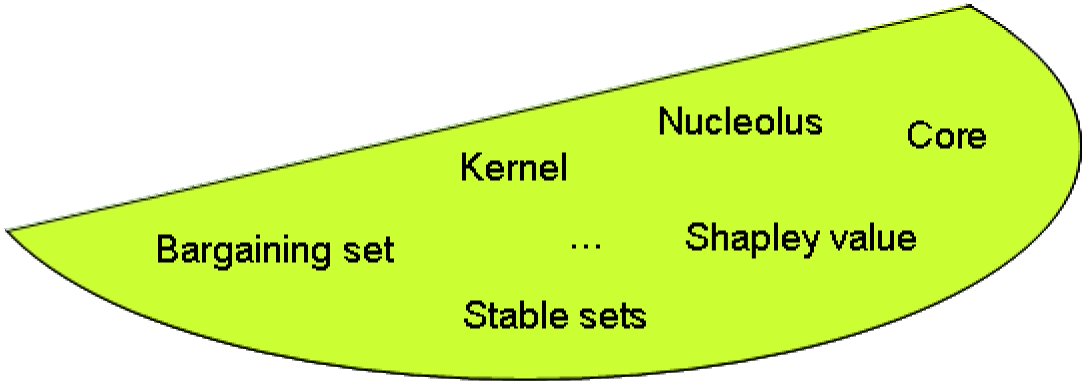


Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

Which actions have to be performed?

Solution Concepts



Cooperative Game Theory(1/2)

To perform some task

Utility distribution, if the task is performed

Jointly perform the task (with some cost)

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**

Cooperative Game Theory (1/2)

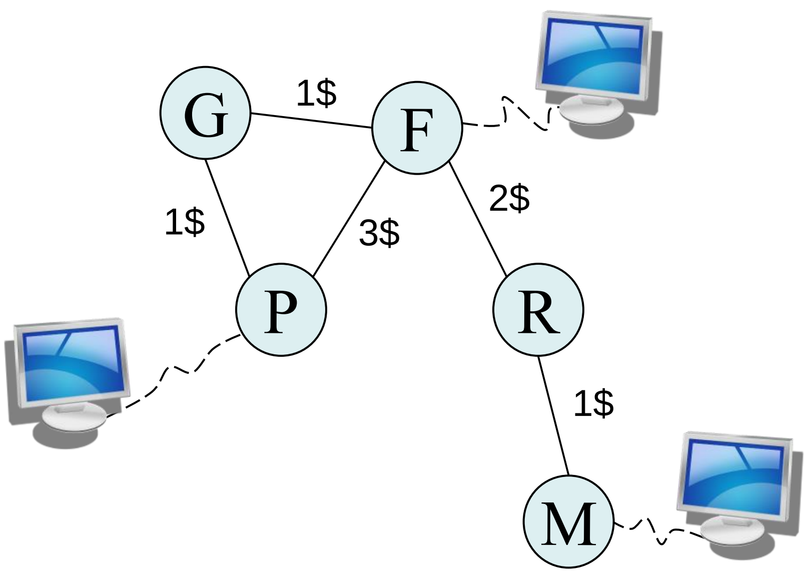
To perform some task

Utility distribution, if the task is performed

Jointly perform the task (with some cost)

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**



- Players get **9$**, if they enforce connectivity
- Enforcing connectivity over an edge as a cost

Cooperative Game Theory$_{(1/2)}$

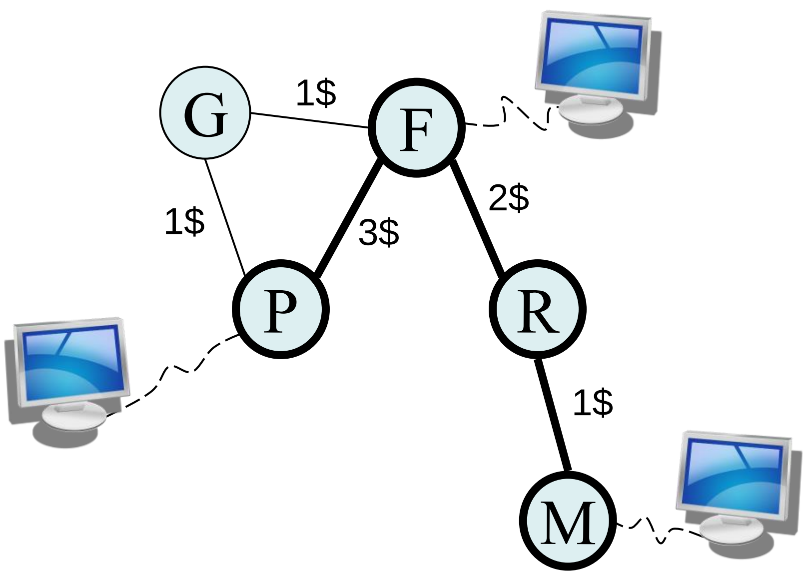
To perform some task

Utility distribution, if the task is performed

Jointly perform the task (with some cost)

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**



- Players get **9$**, if they enforce connectivity
- Enforcing connectivity over an edge as a cost

Coalition {F,P,R,M} gets **9$**, and pays **6$**

worth v({F,P,R,M}) = 9$ - 6$

Cooperative Game Theory$_{(1/2)}$

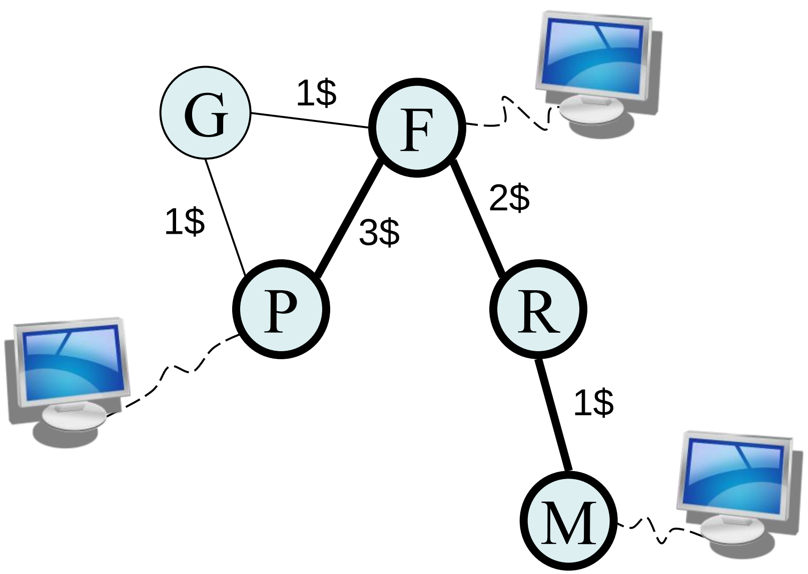
To perform some task

Utility distribution, if the task is performed

Jointly perform the task (with some cost)

Each player:

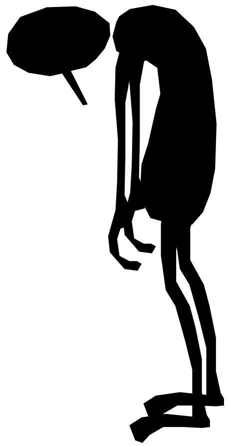
- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**



coalition	worth
{F}	0
…	0
{G,P,R,M}	0
{F,P,R,M}	**3**
{G,F,P,R,M}	4

How to distribute 9$, based on such worths?

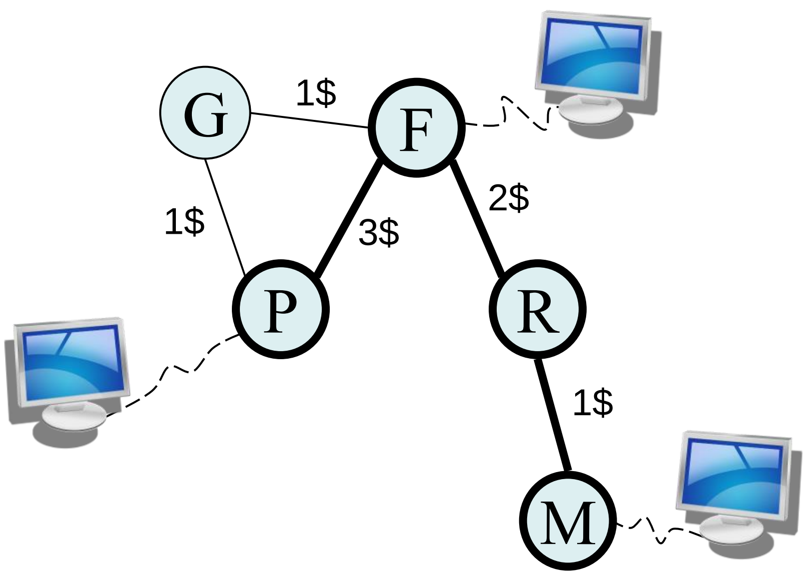
Cooperative Game Theory$_{(2/2)}$



fairness

Each player:

- Has a **goal** to be achieved
- Has a set of possible **actions**
- **Interacts** with other players
- Is **rational**



coalition	worth
{F}	0
…	0
{G,P,R,M}	0
{F,P,R,M}	**3**
{G,F,P,R,M}	4

How to distribute 9$, based on such worths?

The Model

- Players form *coalitions*
- Each coalition is associated with a *worth*
- A *total worth* has to be distributed

$$\mathcal{G} = \langle N, v \rangle,\ v : 2^N \mapsto \mathbb{R}$$

- Outcomes belong to the imputation set $X(\mathcal{G})$

$$x \in X(\mathcal{G}) \quad \begin{cases} \text{Efficiency} \\ x(N) = v(N) \\ \text{Individual Rationality} \\ x_i \geq v(\{i\}), \quad \forall i \in N \end{cases}$$

- *Efficiency*

$$x(N) = v(N)$$

- *Individual Rationality*

$$x_i \geq v(\{i\}), \quad \forall i \in N$$

The Model

- Players form *coalitions*
- Each coalition is associated with a *worth*
- A *total worth* has to be distributed

$$\mathcal{G} = \langle N, v \rangle,\ v : 2^N \mapsto \mathbb{R}$$

- **Solution Concepts** characterize outcomes in terms of
  - Fairness
  - Stability

Excess…

- How fairness/stability can be measured?

$$e(S,x) = v(S) - x(S)$$

- The excess is a measure of the dissatisfaction of S

Excess…

- How fairness/stability can be measured?

$$e(S, x) = v(S) - x(S)$$

- The excess is a measure of the dissatisfaction of S

$$v(\{1\}) = v(\{2\}) = v(\{3\}) = 0$$
$$v(\{1,2\})) = v(\{1,3\}) = v(\{2,3\}) = 1$$
$$v(\{1,2,3\}) = 3$$

Excess…

- How fairness/stability can be measured?

$$e(S, x) = v(S) - x(S)$$

- The excess is a measure of the dissatisfaction of S

$x = (0, 0, 3) \longrightarrow e(\{1, 2\}, x) = v(\{1, 2\}) - (x_1 + x_2) = 1 - 0 = 1$

$x = (1, 2, 0) \longrightarrow e(\{1, 2\}, x) = v(\{1, 2\}) - (x_1 + x_2) = 1 - 3 = -2$

$$v(\{1\}) = v(\{2\}) = v(\{3\}) = 0$$
$$v(\{1, 2\})) = v(\{1, 3\}) = v(\{2, 3\}) = 1$$
$$v(\{1, 2, 3\}) = 3$$

…and the Nucleolus

- Arrange excess values in non-increasing order

…and the Nucleolus

- Arrange excess values in non-increasing order

$x = (1, 2, 0)$ $\qquad\qquad$ $\theta(x) = (0, 0, -1, -1, -2, -2)$

$$v(\{1\}) = v(\{2\}) = v(\{3\}) = 0$$
$$v(\{1, 2\})) = v(\{1, 3\}) = v(\{2, 3\}) = 1$$
$$v(\{1, 2, 3\}) = 3$$

…and the Nucleolus

- Arrange excess values in non-increasing order

Core Imputation

$x = (1, 2, 0)$ $\theta(x) = (0, 0, -1, -1, -2, -2)$

$$v(\{1\}) = v(\{2\}) = v(\{3\}) = 0$$
$$v(\{1, 2\})) = v(\{1, 3\}) = v(\{2, 3\}) = 1$$
$$v(\{1, 2, 3\}) = 3$$

…and the Nucleolus

- Arrange excess values in non-increasing order

$$x^* = (1, 1, 1) \qquad \theta(x^*) = (-1, -1, -1, -1, -1, -1)$$

$$\boxed{x = (1, 2, 0) \qquad \theta(x) = (0, 0, -1, -1, -2, -2)}$$

$$v(\{1\}) = v(\{2\}) = v(\{3\}) = 0$$
$$v(\{1, 2\})) = v(\{1, 3\}) = v(\{2, 3\}) = 1$$
$$v(\{1, 2, 3\}) = 3$$

…and the Nucleolus

- Arrange excess values in non-increasing order

$$\boxed{x^* = (1,1,1) \qquad \theta(x^*) = (-1,-1,-1,-1,-1,-1)}$$

$$x = (1,2,0) \qquad \theta(x) = (0,0,-1,-1,-2,-2)$$

$$v(\{1\}) = v(\{2\}) = v(\{3\}) = 0$$
$$v(\{1,2\})) = v(\{1,3\}) = v(\{2,3\}) = 1$$
$$v(\{1,2,3\}) = 3$$

…and the Nucleolus

- Arrange excess values in non-increasing order

Definition [Schmeidler]

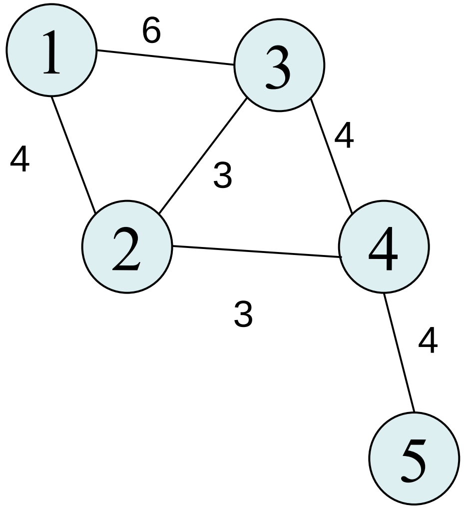
The *nucleolus* $\mathcal{N}(\mathcal{G})$ of a game $\mathcal{G}$ is the set
$\mathcal{N}(\mathcal{G}) = \{x \in X(\mathcal{G}) \mid \nexists y \in X(\mathcal{G}) \text{ s.t. } \theta(y) \prec \theta(x)\}$

$$\boxed{x^* = (1, 1, 1) \qquad \theta(x^*) = (-1, -1, -1, -1, -1, -1)}$$

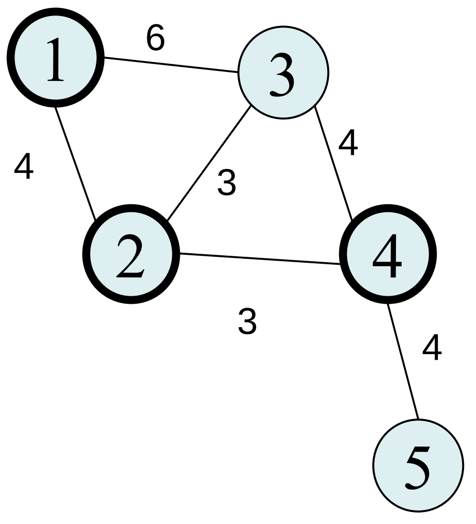
$$x = (1, 2, 0) \qquad \theta(x) = (0, 0, -1, -1, -2, -2)$$

$$v(\{1\}) = v(\{2\}) = v(\{3\}) = 0$$
$$v(\{1, 2\})) = v(\{1, 3\}) = v(\{2, 3\}) = 1$$
$$v(\{1, 2, 3\}) = 3$$

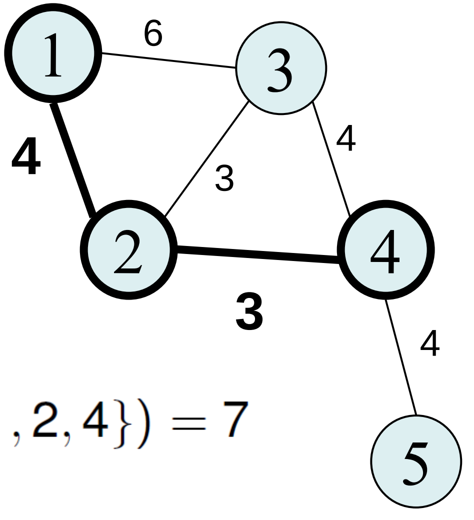
Graph Games



Graph Games

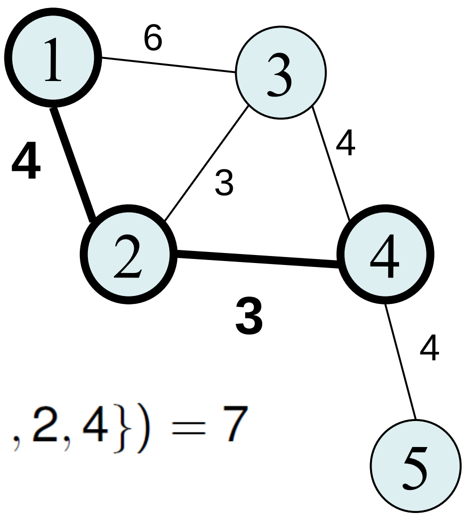


Graph Games



$v(\{1,2,4\}) = 7$

Graph Games

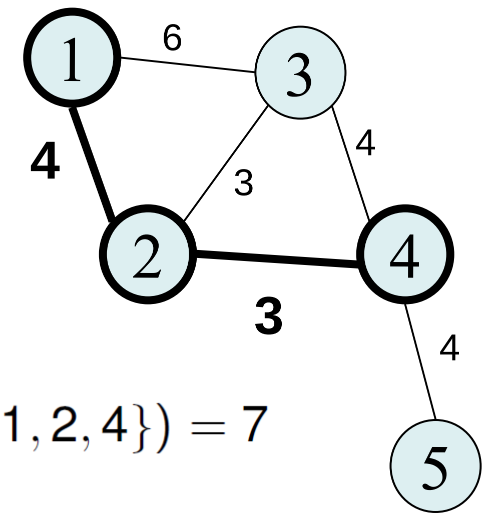


$v(\{1,2,4\}) = 7$

- *Graph Games* [Deng and Papadimitriou, 1994]
  - Computational issues of several solution concepts
  - The *(pre)nucleolus* can be computed in **P**

$$x_i^* = \frac{1}{2}\sum_{j \neq i} w_{i,j}$$

Graph Games



$v(\{1,2,4\}) = 7$

- *Graph Games* [Deng and Papadimitriou, 1994]
  - Computational issues of several solution concepts
  - The *(pre)nucleolus* can be computed in **P**

$$x_i^* = \frac{1}{2}\sum_{j \neq i} w_{i,j}$$

- *Cost allocation on trees* [Megiddo, 1978]
  - Polynomial time algorithm
- *Flow games* [Deng, Fang, and Sun, 2006]
  - Polynomial time algorithm on simple networks (unitary edge capacity)
  - **NP**-hard, in general
- *Weighted voting games* [Elkind and Pasechnik, 2009]
  - Pseudopolynomial algorithm

Kopelowitz, 1967

$$\mathtt{LP}_1 \begin{cases} \min \epsilon_1 & \\ \quad e(S,x) \leq \epsilon_1 & \forall S \subset N, S \notin W_0 = \{\varnothing\} \\ \quad x \in X(\mathcal{G}) & \end{cases}$$

Kopelowitz, 1967

$$\mathrm{LP}_1 \begin{cases} \min \epsilon_1 & \\ e(S,x) \leq \epsilon_1 & \forall S \subset N, S \notin W_0 = \{\varnothing\} \\ x \in X(\mathcal{G}) & \end{cases}$$

$$\mathrm{LP}_2 \begin{cases} \min \epsilon_2 & \\ e(S,x) = \epsilon_1^* & \forall S \in W_1 \\ e(S,x) \leq \epsilon_2 & \forall S \subset N, S \notin (W_0 \cup W_1) \\ x \in X(\mathcal{G}) & \end{cases}$$

where:

- $V_1 = \{x \mid (x, \epsilon_1^*)$ is an optimal solution to $\mathrm{LP}_1\}$
- $W_1 = \{S \subseteq N \mid e(S,x) = \epsilon_1^*,$ for every $x \in V_1\}$

Kopelowitz, 1967

$$\mathtt{LP}_k \left\{ \begin{array}{ll} \min \epsilon_k & \\ \quad e(S,x) = \epsilon_r^* & \forall S \in W_r, r \in \{1,\ldots,k-1\} \\ \quad e(S,x) \le \epsilon_k & \forall S \subset N, S \notin (W_0 \cup \cdots \cup W_{k-1}) \\ \quad x \in X(\mathcal{G}) & \end{array} \right.$$

where:

- $V_r = \{x \mid (x, \epsilon_r^*) \text{ is an optimal solution to } \mathtt{LP}_r\}$
- $W_r = \{S \subseteq N \mid e(S,x) = \epsilon_r^*, \text{ for every } x \in V_r\}$

How many iterations?

An Example Computation

$N = 1, ..., n, n+1, n+2$

$v(N) = n + 2$
$v(\{i\}) = 1,\ i \in \{1, ..., n\}$
$v(\{1, ..., n\}) = n$
$v(\{n+1\}) = v(\{n+2\}) = 0$
$v(\{n+1, n+2\}) = 2$
$v(S) = -\infty,\ |\{n+1, n+2\} \cap S| \geq 1,$
$|\{1, ..., n\} \cap S| \geq 1,\ S \neq N$

$S_1, S_2, ... \subset \{1, ..., n\}$ $|S_i| > 1$
$v(S_i) = |S_i| - 1 + 2^{-i}$

An Example Computation

$N = 1, ..., n, n+1, n+2$

$v(N) = n+2$
$v(\{i\}) = 1,\ i \in \{1, ..., n\}$
$v(\{1, ..., n\}) = n$
$v(\{n+1\}) = v(\{n+2\}) = 0$
$v(\{n+1, n+2\}) = 2$
$v(S) = -\infty,\ |\{n+1, n+2\} \cap S| \geq 1,$
$\quad |\{1, ..., n\} \cap S| \geq 1,\ S \neq N$

$S_1, S_2, ... \subset \{1, ..., n\}\ \ |S_i| > 1$
$v(S_i) = |S_i| - 1 + 2^{-i}$

$\epsilon_1^* = 0$

$x^* = (1, \ldots, 1, x_{n+1}^*, x_{n+2}^*)$

$$\mathrm{LP}_1 \left\{ \begin{array}{l} \min \epsilon_1 \\ \quad n - x(\{1, ..., n\}) \leq \epsilon_1 \\ \quad 2 - x_{n+1} - x_{n+2} \leq \epsilon_1 \\ \quad x(\{1, ..., n\}) + x_{n+1} + x_{n+2} = n+2 \\ \quad x_i \geq 1,\ i \in \{1, ..., n\} \\ \quad \vdots \end{array} \right.$$

An Example Computation

$N = 1, ..., n, n+1, n+2$

$$v(N) = n+2$$
$$v(\{i\}) = 1,\ i \in \{1, ..., n\}$$
$$v(\{1, ..., n\}) = n$$
$$v(\{n+1\}) = v(\{n+2\}) = 0$$
$$v(\{n+1, n+2\}) = 2$$
$$v(S) = -\infty,\ |\{n+1, n+2\} \cap S| \geq 1,\ |\{1, ..., n\} \cap S| \geq 1,\ S \neq N$$

$S_1, S_2, ... \subset \{1, ..., n\}$ $|S_i| > 1$

$$v(S_i) = |S_i| - 1 + 2^{-i}$$

$$\epsilon_1^* = 0$$

$$x^* = (1, \ldots, 1, x_{n+1}^*, x_{n+2}^*)$$

The excess is constant

$$e(S_i, x^*) = v(S_i) - x^*(S_i) = -1 + 2^{-i}$$

An Example Computation

$N = 1, ..., n, n+1, n+2$

$v(N) = n+2$
$v(\{i\}) = 1,\ i \in \{1, ..., n\}$
$v(\{1, ..., n\}) = n$
$v(\{n+1\}) = v(\{n+2\}) = 0$
$v(\{n+1, n+2\}) = 2$
$v(S) = -\infty,\ |\{n+1, n+2\} \cap S| \geq 1,$
$|\{1, ..., n\} \cap S| \geq 1,\ S \neq N$

$S_1, S_2, ... \subset \{1, ..., n\}\quad |S_i| > 1$
$v(S_i) = |S_i| - 1 + 2^{-i}$

$\epsilon_1^* = 0$

$x^* = (1, \ldots, 1, x_{n+1}^*, x_{n+2}^*)$

The excess is constant

$e(S_i, x^*) = v(S_i) - x^*(S_i) = -1 + 2^{-i}$

$e(S_i, x^*) \leq \epsilon_2$
$\epsilon_2^* = -1 + 2^{-1}$

An Example Computation

$$N = 1, \ldots, n, n+1, n+2$$

$$v(N) = n+2$$
$$v(\{i\}) = 1,\ i \in \{1, \ldots, n\}$$
$$v(\{1, \ldots, n\}) = n$$
$$v(\{n+1\}) = v(\{n+2\}) = 0$$
$$v(\{n+1, n+2\}) = 2$$
$$v(S) = -\infty,\ |\{n+1, n+2\} \cap S| \geq 1,\ |\{1, \ldots, n\} \cap S| \geq 1,\ S \neq N$$

$$S_1, S_2, \ldots \subset \{1, \ldots, n\} \quad |S_i| > 1$$
$$v(S_i) = |S_i| - 1 + 2^{-i}$$

$$\epsilon_1^* = 0$$
$$x^* = (1, \ldots, 1, x_{n+1}^*, x_{n+2}^*)$$

The excess is constant

$$e(S_i, x^*) = v(S_i) - x^*(S_i) = -1 + 2^{-i}$$

$$e(S_i, x^*) \leq \epsilon_3$$
$$\epsilon_2^* = -1 + 2^{-1} > \epsilon_3^* = -1 + 2^{-2}, \ldots >$$

Kopelowitz, 1967

$$\mathrm{LP}_k \left\{ \begin{array}{ll} \min \epsilon_k & \\ \quad e(S,x) = \epsilon_r^* & \forall S \in W_r, r \in \{1,\ldots,k-1\} \\ \quad e(S,x) \le \epsilon_k & \forall S \subset N, S \notin (W_0 \cup \cdots \cup W_{k-1}) \\ \quad x \in X(\mathcal{G}) & \end{array} \right.$$

where:

- $V_r = \{x \mid (x, \epsilon_r^*) \text{ is an optimal solution to } \mathrm{LP}_r\}$
- $W_r = \{S \subseteq N \mid e(S,x) = \epsilon_r^*, \text{ for every } x \in V_r\}$

How many iterations?

Kopelowitz, 1967

$$\mathtt{LP}_k \left\{ \begin{array}{ll} \min \epsilon_k & \\ \quad e(S,x) = \epsilon_r^* & \forall S \in W_r, r \in \{1,\ldots,k-1\} \\ \quad e(S,x) \leq \epsilon_k & \forall S \subset N, S \notin (W_0 \cup \cdots \cup W_{k-1}) \\ \quad x \in X(\mathcal{G}) & \end{array} \right.$$

where:

- $V_r = \{x \mid (x, \epsilon_r^*) \text{ is an optimal solution to } \mathtt{LP}_r\}$
- $W_r = \{S \subseteq N \mid e(S,x) = \epsilon_r^*, \text{ for every } x \in V_r\}$

Theorem

The algorithm performs $\Omega(2^n)$ steps, in some cases.

cf. Mashler, Peleg, and Shapley, 1979

$$
\mathtt{LP}_k \begin{cases} \min \epsilon_k \\ e(S,x) = \epsilon_r^* & \forall S \in W_r, r \in \{1,\ldots,k-1\} \\ e(S,x) \leq \epsilon_k & \forall S \subset N, S \notin (W_0 \cup \cdots \cup W_{k-1}) \\ x \in X(\mathcal{G}) \end{cases}
$$

where:

- $V_r = \{x \mid (x, \epsilon_r^*) \text{ is an optimal solution to } \mathtt{LP}_r\}$
- $W_r = \{S \subseteq N \mid e(S,x) = \epsilon_r^*, \text{ for every } x \in V_r\}$

cf. Mashler, Peleg, and Shapley, 1979

$$\mathrm{LP}_k \left\{ \begin{array}{ll} \min \epsilon_k & \\ e(S,x) = \epsilon_r^* & \forall S \in W_r, r \in \{1,\ldots,k-1\} \\ e(S,x) \le \epsilon_k & \forall S \subset N, S \notin \cancel{(W_0 \cup \ldots \cup W_{k-1})} \\ x \in X(\mathcal{G}) & \end{array} \right.$$

where:

- $V_r = \{x \mid (x, \epsilon_r^*) \text{ is an optimal solution to } \mathrm{LP}_r\}$
- $W_r = \{S \subseteq N \mid e(S,x) = \epsilon_r^*, \text{ for every } x \in V_r\}$

$$\boxed{\{S \subseteq N \mid x(S) = y(S), \forall x, y \in V_{k-1}\}}$$

cf. Mashler, Peleg, and Shapley, 1979

$$
\mathrm{LP}_k \begin{cases}
\min \epsilon_k \\
e(S,x) = \epsilon_r^* & \forall S \in W_r,\, r \in \{1,\ldots,k-1\} \\
e(S,x) \leq \epsilon_k & \forall S \subset N,\, S \notin \cancel{(W_0 \cup \cdots \cup W_{k-1})} \\
x \in X(\mathcal{G})
\end{cases}
$$

where:

- $V_r = \{x \mid (x, \epsilon_r^*) \text{ is an optimal solution to } \mathrm{LP}_r\}$
- $W_r = \{S \subseteq N \mid e(S,x) = \epsilon_r^*, \text{ for every } x \in V_r\}$

$$\boxed{\{S \subseteq N \mid x(S) = y(S), \forall x, y \in V_{k-1}\}}$$

[Kern and Paulusuma, 2003]

LP Approaches over Compact Games

$$\mathrm{LP}_k \begin{cases} \min \epsilon_k \\ \quad e(S,x) = \epsilon_r^* \qquad \forall S \in W_r, r \in \{1,\ldots,k-1\} \\ \quad e(S,x) \leq \epsilon_k \qquad \forall S \subset N, S \notin \mathcal{F}_{k-1} \\ \quad x \in X(\mathcal{G}) \end{cases}$$

where:

- $V_r = \{x \mid (x, \epsilon_r^*) \text{ is an optimal solution to } \mathrm{LP}_r\}$
- $W_r = \{S \subseteq N \mid e(S,x) = \epsilon_r^*, \text{ for every } x \in V_r\}$
- $\mathcal{F}_{k-1} = \{S \subseteq N \mid x(S) = y(S), \forall x, y \in V_{k-1}\}$

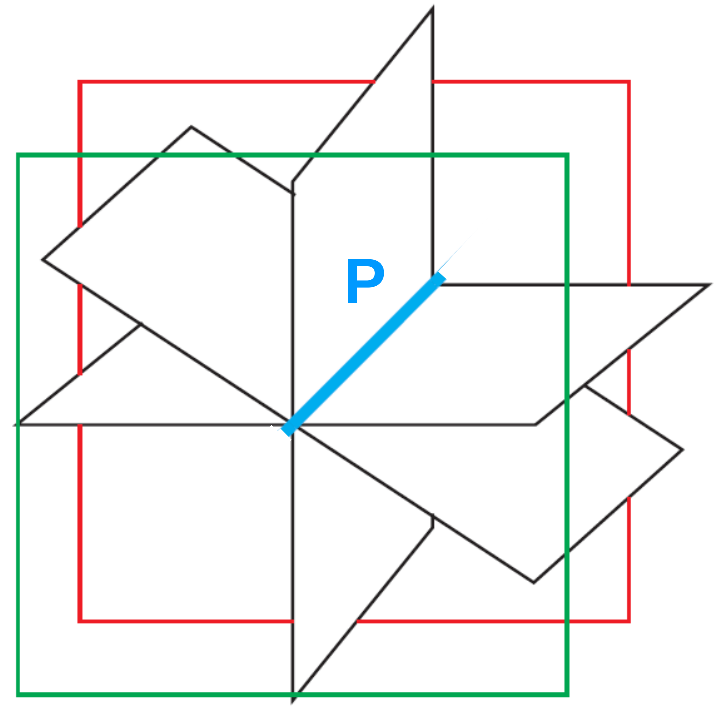
- In compact games, two problems have to be faced:
  - **(P1)** Sets W and $\mathcal{F}$ contain exponentially many elements, but we would like to avoid listing them explicitly
  - **(P2)** Translate LP (complexity) results to "succinct programs"

(P1): A Convenient Representation

$W \qquad \mathcal{F}$

fixed inequalities

equalities + implied equalities

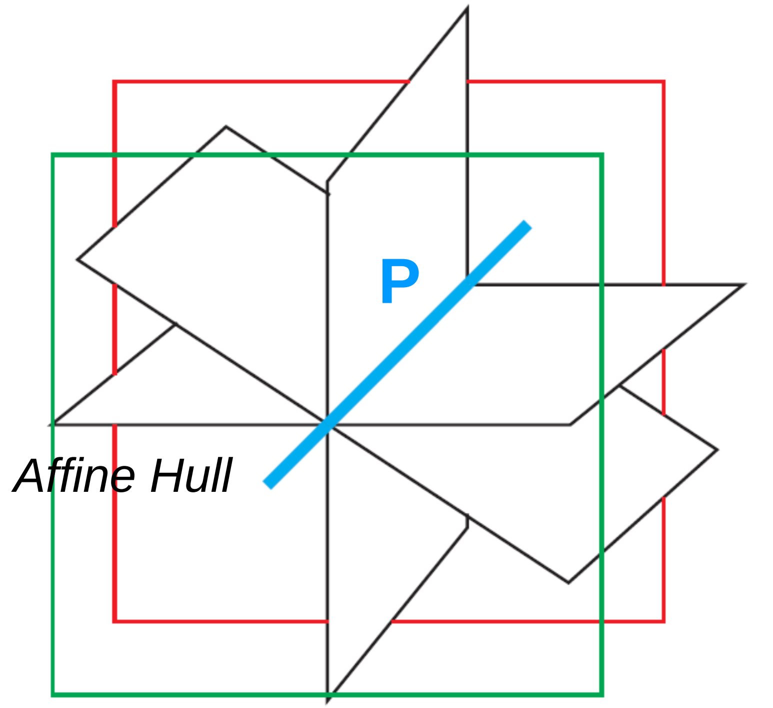


(P1): A Convenient Representation

W $\mathcal{F}$

fixed inequalities

equalities + implied equalities



Theorem

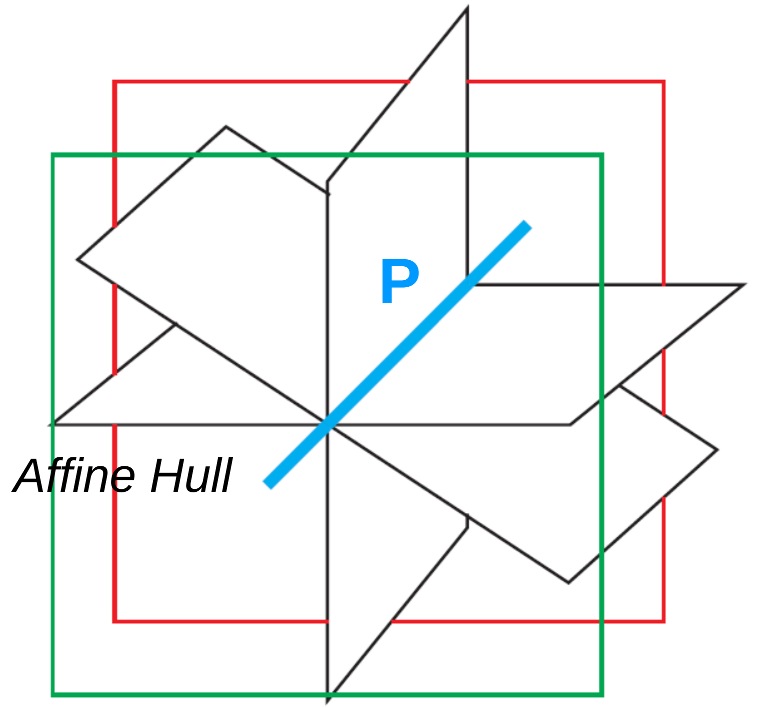
- $aff.hull(V_k) =$ *solutions for equalities over* $W_k \cup W_{k-1} \cup \cdots \cup W_1$

(P1): A Convenient Representation

$W \qquad \mathcal{F}$

fixed inequalities

equalities + implied equalities



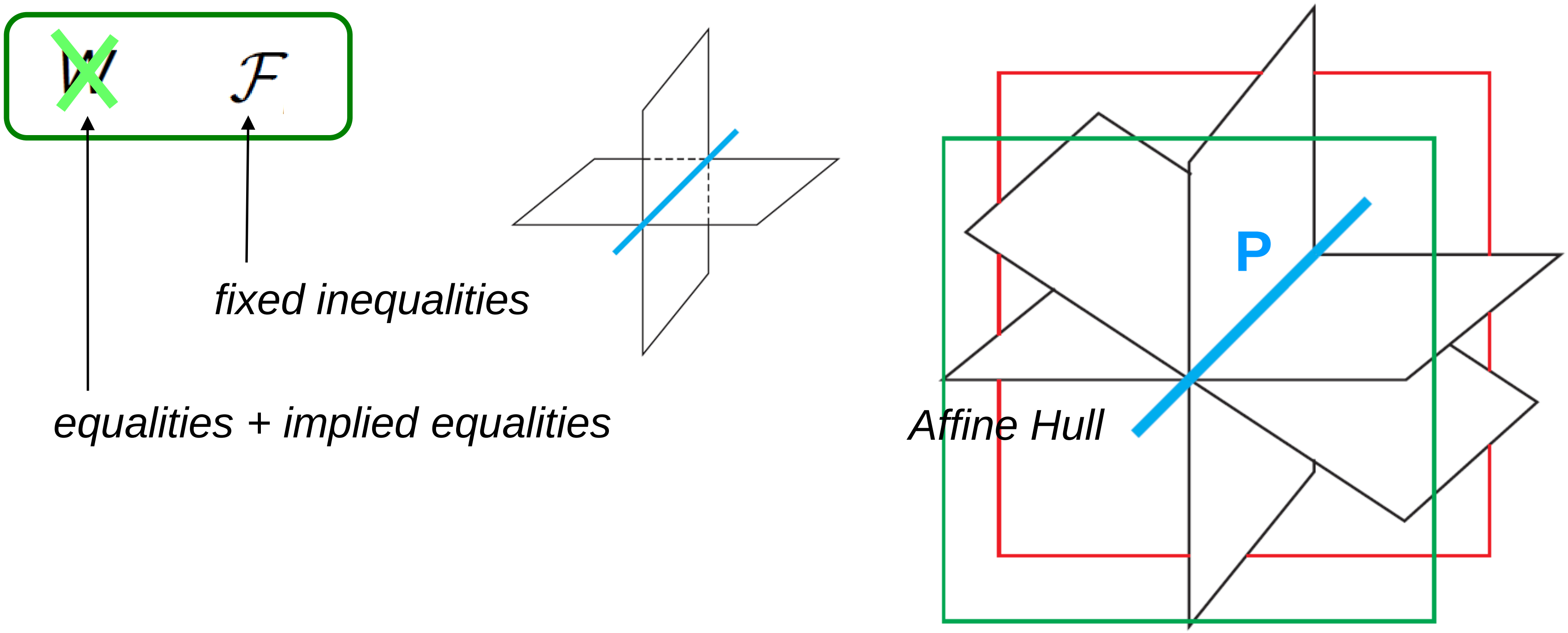
Theorem

- $aff.hull(V_k) = $ *solutions for equalities over* $W_k \cup \underbrace{W_{k-1} \cup \cdots \cup W_1}_{\textit{equalities}}$

  $\{S \subseteq N \mid e(S, x) = \epsilon_k^*, \text{ for every } x \in V_k\}$

  Implied equalities

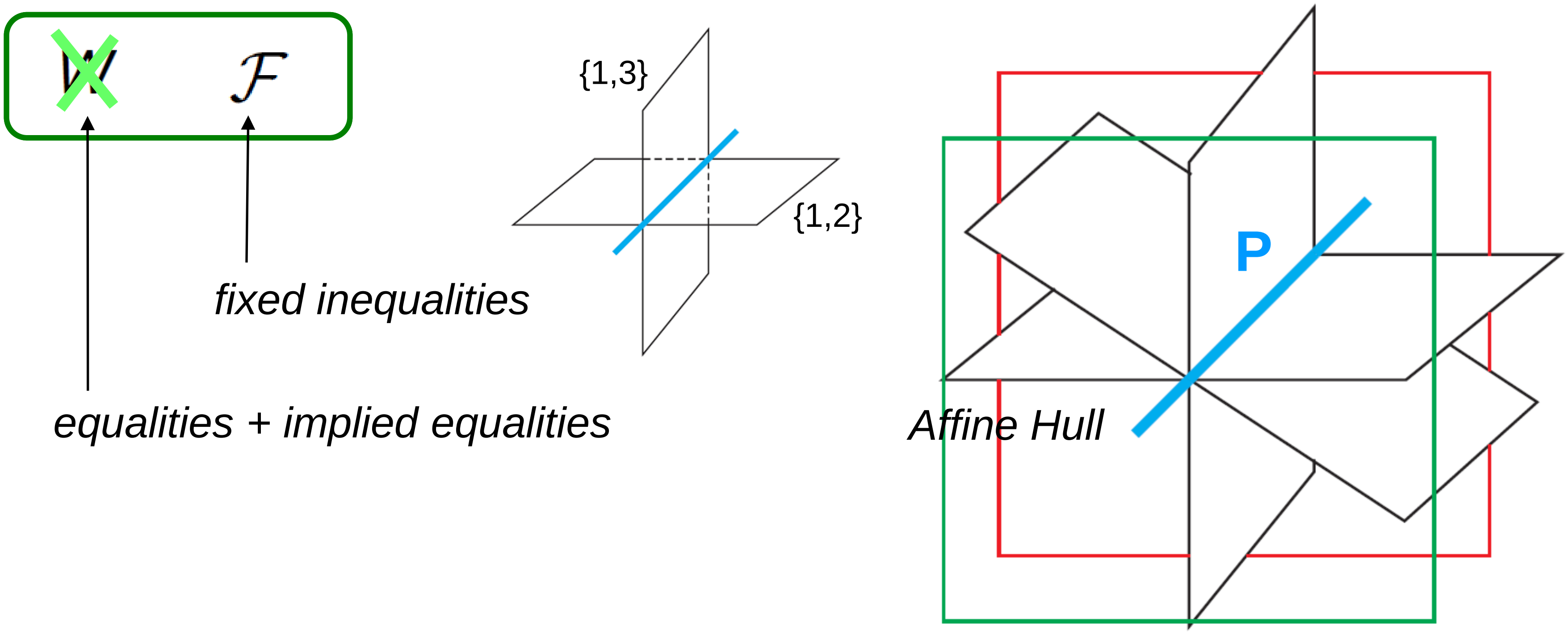
(P1): A Convenient Representation



Theorem

- $aff.hull(V_k) = $ *solutions for equalities over* $W_k \cup W_{k-1} \cup \cdots \cup W_1$
- *A basis* $\mathcal{B}_k$ *for* $aff.hull(V_k)$ *contains n vectors at most*

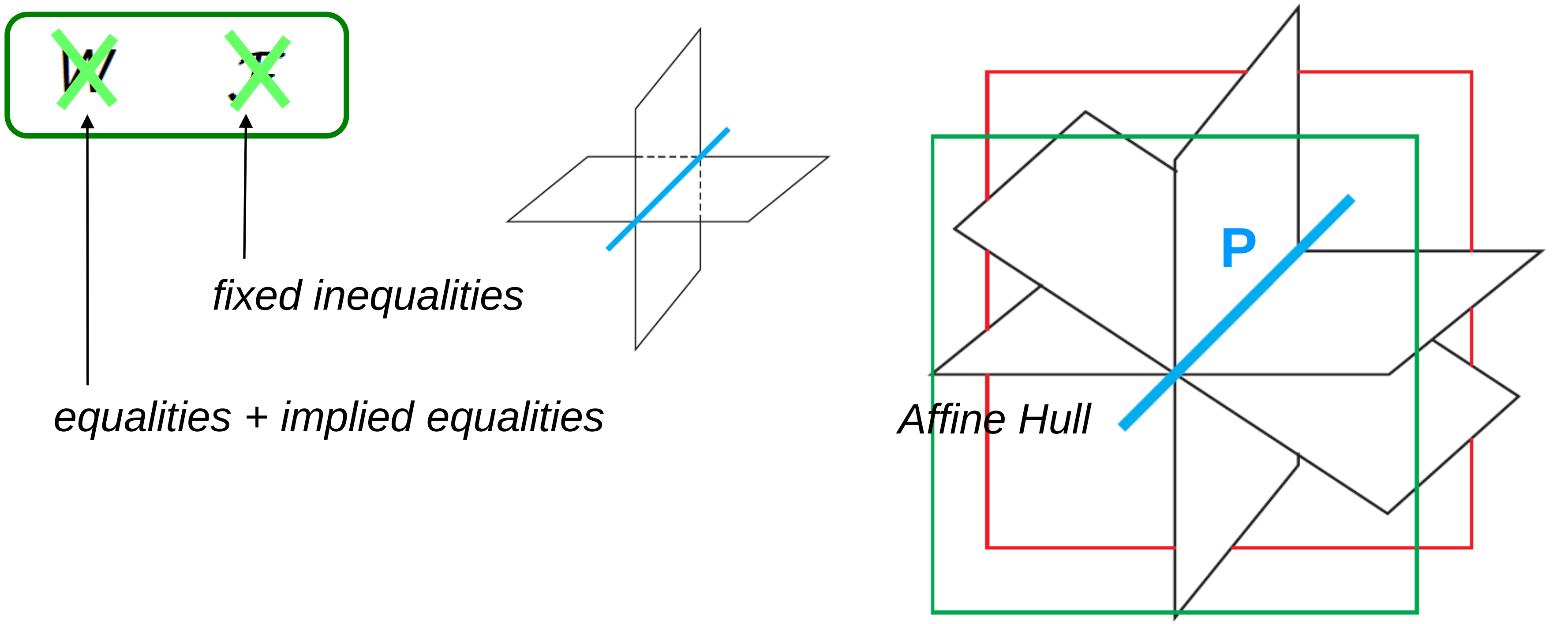
(P1): A Convenient Representation



Theorem

- *$aff.hull(V_k)$ = solutions for equalities over $W_k \cup W_{k-1} \cup \cdots \cup W_1$*
- *A basis $\mathcal{B}_k$ for $aff.hull(V_k)$ contains n vectors at most*
- *$S \in \mathcal{F}_k$ iff S is a linear combination of the* indicator *vectors for $\mathcal{B}_k$*

(P1): A Convenient Representation



Theorem

- *$aff.hull(V_k)$ = solutions for equalities over $W_k \cup W_{k-1} \cup \cdots \cup W_1$*
- *A basis $\mathcal{B}_k$ for $aff.hull(V_k)$ contains n vectors at most*
- *$S \in \mathcal{F}_k$ iff S is a linear combination of the* indicator *vectors for $\mathcal{B}_k$*

(P1): A Convenient Representation

$$v : 2^N \mapsto \mathbb{R}$$

$$+$$

$$aff.hull$$

$$\Longrightarrow \quad \mathrm{LP}_{k+1}$$

$$\xrightarrow{\ i\ } \boxed{\mathrm{LP}} \xrightarrow{\ \textit{i-th inequality}\ }$$

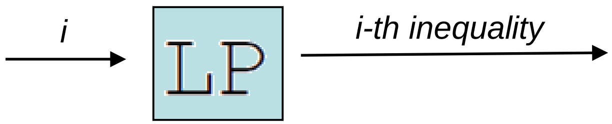
Theorem

- $aff.hull(V_k)$ = solutions for equalities over $W_k \cup W_{k-1} \cup \cdots \cup W_1$
- A basis $\mathcal{B}_k$ for $aff.hull(V_k)$ contains n vectors at most
- $S \in \mathcal{F}_k$ iff S is a linear combination of the indicator vectors for $\mathcal{B}_k$

(P2) Computation Problems

- In compact games, two problems have to be faced:
  - **(P1)** Sets W and $\mathcal{F}$ contain exponentially many elements, but we would like to avoid listing them explicitly
  - **(P2)** Translate LP (complexity) results to "succinct programs"

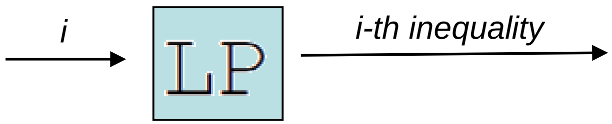
(P2) Computation Problems



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

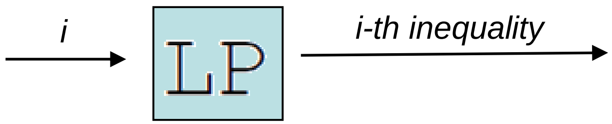
- In compact games, two problems have to be faced:
  - **(P1)** Sets W and $\mathcal{F}$ contain exponentially many elements, but we would like to avoid listing them explicitly
  - **(P2)** Translate LP (complexity) results to “succinct programs”

Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

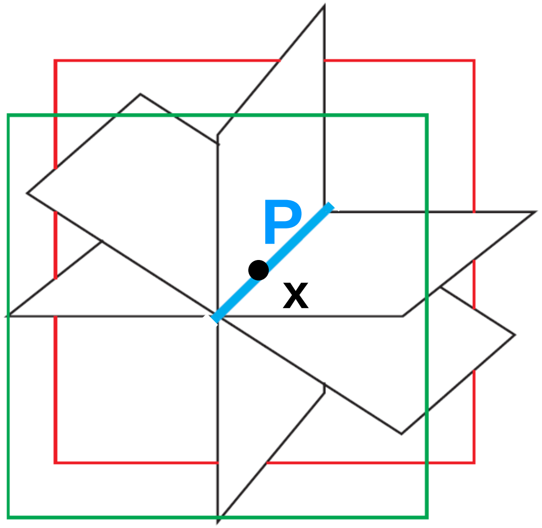
Complexity Results



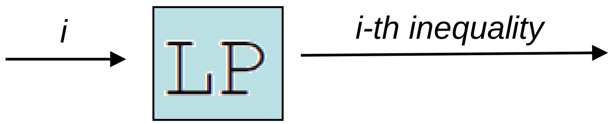
Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

Trivial

- Given a vector **x**, we can:
  - Guess an index *i*
  - Check that the *i-th inequality* is not satisfied by **x**

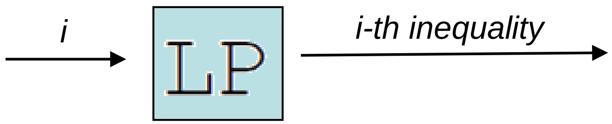


Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta}_2^P$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$

Complexity Results

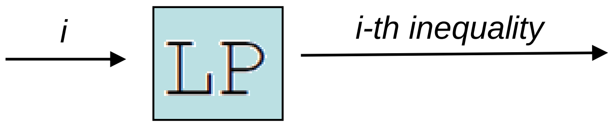


Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

Proof

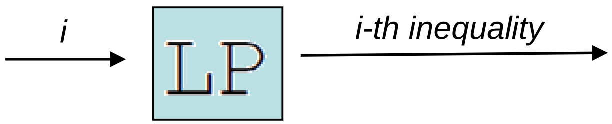
- By **Helly's theorem**, we can solve the complementary problem in **NP**:
  - Guess **n+1** inequalities
  - Check that they are not satisfiable (in polynomial time)

Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta}_2^P$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$

Proof Overview

(1) The dimension is **n-k** at most, if there are at least **k** linear independent implied equalities

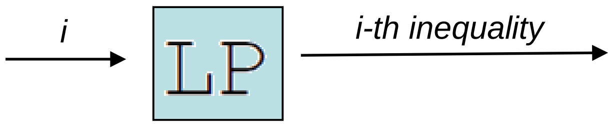
(2) In order to check that the *i-th* inequality is an implied one,

we can guess in **NP** a **support set** *W(i),* again by Helly's theorem:

- **n** inequalities + the *i-th* inequality treated as strict
- *W(i)* is not satisfiable, which can be checked in polynomial time

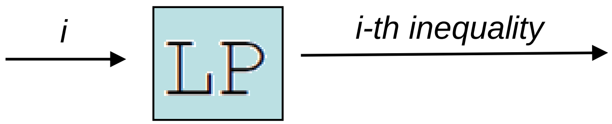
- Guess **k** implied equalities plus their support sets
- Check that they are linear independent

Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F}\boldsymbol{\Delta}_2^P$
OPTIMALVALUECOMPUTATION	in $\mathbf{F}\boldsymbol{\Delta}_2^P$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F}\boldsymbol{\Delta}_2^P$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F}\boldsymbol{\Delta}_2^P$

Complexity Results

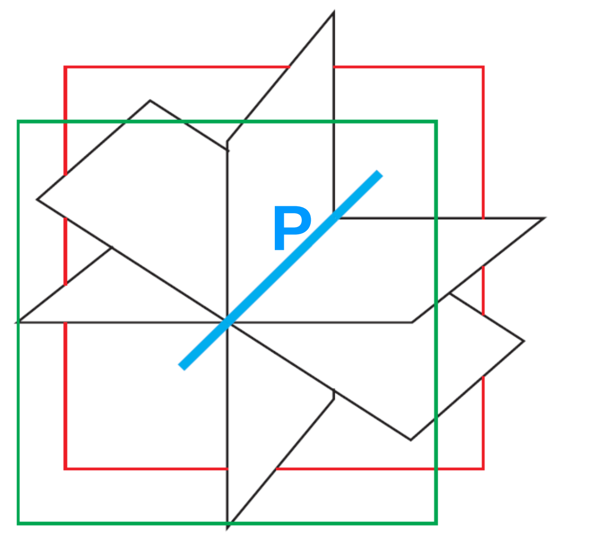


Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

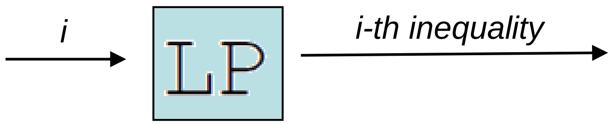
Proof

(1) Compute the dimension **n-k**, with a *binary search* invoking an **NP** oracle

(2) Guess **k** implied equalities plus their support sets

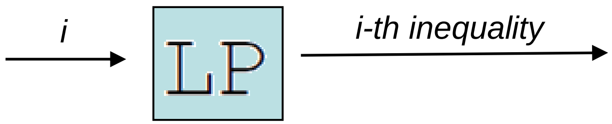


Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta}_2^P$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$

Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta}_2^P$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$

Routine

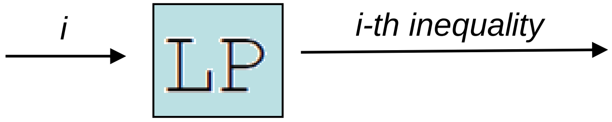
(1) Bfs can be represented with polynomially many bits

(2) LP induces a polytope and hence the optimum is achieved on some bfs.

(3) Perform a *binary search* over the range of the optimum solution:

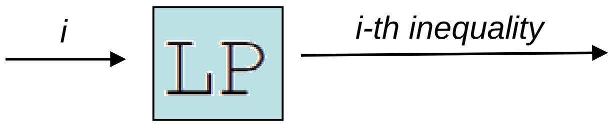
- Add the current value as a constraint, and check satisfiability

Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

Complexity Results

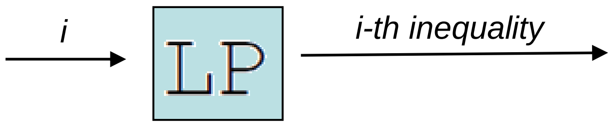


Problem	**Result**
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta}_2^P$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$

Routine

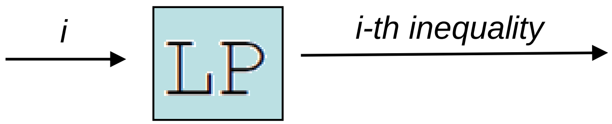
- LP induces a polytope
- Compute the lexicographically maximum bfs solution, by iterating over the various components, and treating each of them as an objective function to be optimized.

Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta}_2^P$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$

Complexity Results



Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

Routine

(1) Compute the optimum value

(2) Define LP’ as LP plus the constraint stating that the objective function must equal the optimum value

(3) Compute a feasible value for LP’

Putting It All Togheter

$$
\mathrm{LP}_k \begin{cases} \min \epsilon_k \\ e(S,x) = \epsilon^*_r & \forall S \in W_r, r \in \{1,\ldots,k-1\} \\ e(S,x) \leq \epsilon_k & \forall S \subset N, S \notin \mathcal{F}_{k-1} \\ x \in X(\mathcal{G}) \\ \textit{where:} \\ \bullet\; V_r = \{x \mid (x,\epsilon^*_r) \text{ is an optimal solution to } \mathrm{LP}_r\} \\ \bullet\; W_r = \{S \subseteq N \mid e(S,x) = \epsilon^*_r, \text{ for every } x \in V_r\} \\ \bullet\; \mathcal{F}_{k-1} = \{S \subseteq N \mid x(S) = y(S), \forall x,y \in V_{k-1}\} \end{cases}
$$

$v : 2^N \mapsto \mathbb{R}$ + *aff.hull* $\longrightarrow \mathrm{LP}_{k+1}$

Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta}_2^P$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta}_2^P$

M.P.S. **Compact Encoding** **Algorithms in $\mathbf{F\Delta_2^P}$**

- In compact games, two problems have to be faced:
  - **(P1)** Sets W and $\mathcal{F}$ contain exponentially many elements, but we would like to avoid listing them explicitly
  - **(P2)** Translate LP (complexity) results to “succinct programs”

Putting It All Togheter

$$
\mathrm{LP}_k \begin{cases} \min \epsilon_k \\ e(S,x) = \epsilon_r^* & \forall S \in W_r, r \in \{1,\ldots,k-1\} \\ e(S,x) \le \epsilon_k & \forall S \subset N, S \notin \mathcal{F}_{k-1} \\ x \in X(\mathcal{G}) \end{cases}
$$

where:

- $V_r = \{x \mid (x, \epsilon_r^*)$ is an optimal solution to $\mathrm{LP}_r\}$
- $W_r = \{S \subseteq N \mid e(S,x) = \epsilon_r^*,$ for every $x \in V_r\}$
- $\mathcal{F}_{k-1} = \{S \subseteq N \mid x(S) = y(S), \forall x, y \in V_{k-1}\}$

$v : 2^N \mapsto \mathbb{R}$

\+ *aff.hull* $\longrightarrow \mathrm{LP}_{k+1}$

Problem	Result
MEMBERSHIP	in co-**NP**
NONEMPTINESS	in co-**NP**
DIMENSION	in **NP**
AFFINEHULLCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVALUECOMPUTATION	in $\mathbf{F\Delta_2^P}$
FEASIBLEVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$
OPTIMALVECTORCOMPUTATION	in $\mathbf{F\Delta_2^P}$

M.P.S. **Compact Encoding** **Algorithms in $\mathrm{F}\Delta^P_2$**

Theorem

Computing the nucleolus is feasible in $\mathbf{F\Delta_2^P}$. Thus, deciding whether an imputation is the nucleolus is feasible in $\mathbf{\Delta_2^P}$.

Checking Problem

Theorem

Deciding whether an imputation is the nucleolus is $\mathbf{\Delta_2^P}$*-hard. Thus, it is* $\mathbf{\Delta_2^P}$*-complete.*

Checking Problem

Theorem

Deciding whether an imputation is the nucleolus is $\mathbf{\Delta_2^P}$-hard. Thus, it is $\mathbf{\Delta_2^P}$-complete.

Proof (Reduction for Graph Games: *The cost of individual rationality!*)

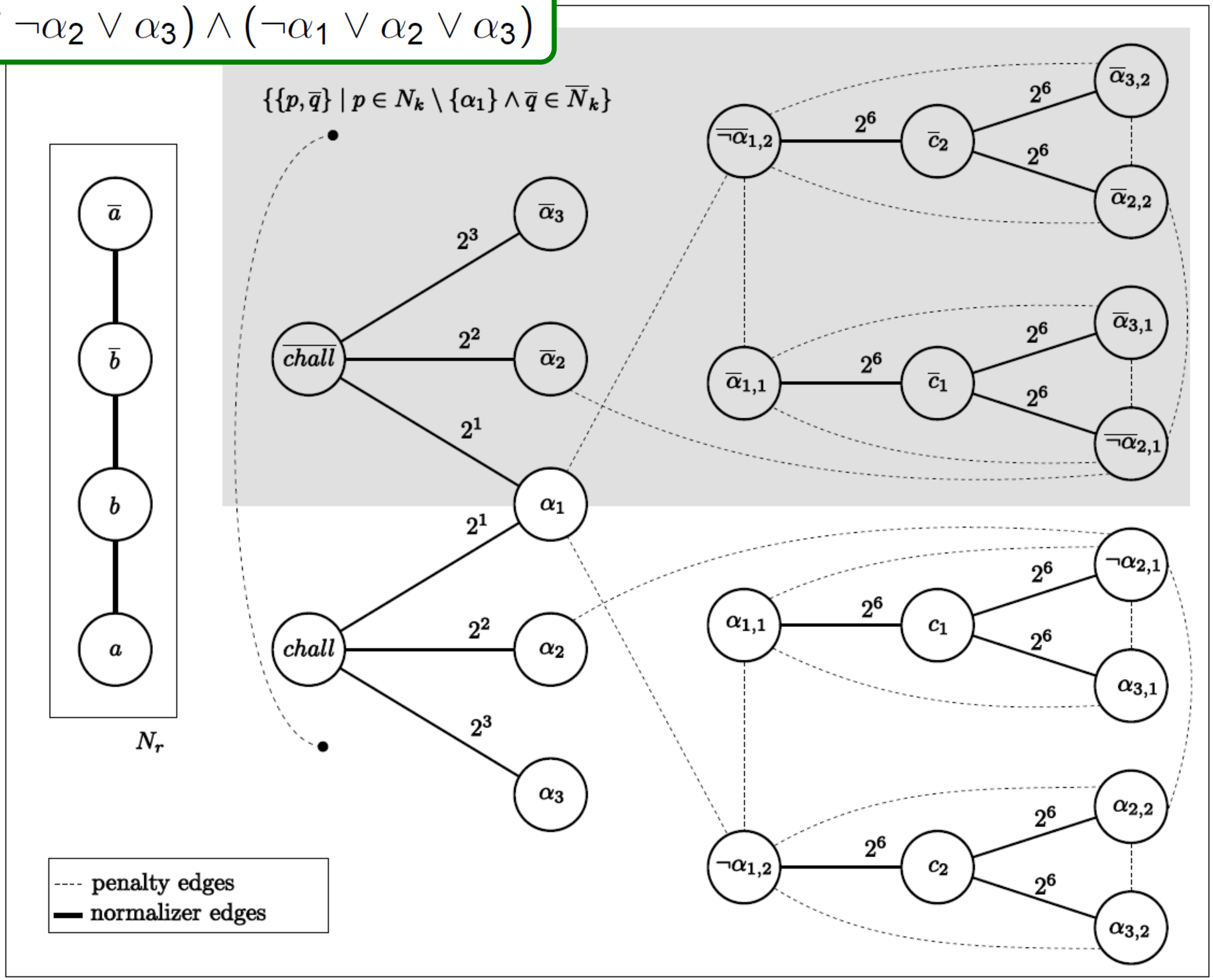
- Deciding the truth value of the least significant variable in the lexicographically maximum satisfying assignment

$$\boxed{\hat{\phi} = (\alpha_1 \vee \neg\alpha_2 \vee \alpha_3) \wedge (\neg\alpha_1 \vee \alpha_2 \vee \alpha_3)}$$

$\alpha_1 < \alpha_2 < \alpha_3$

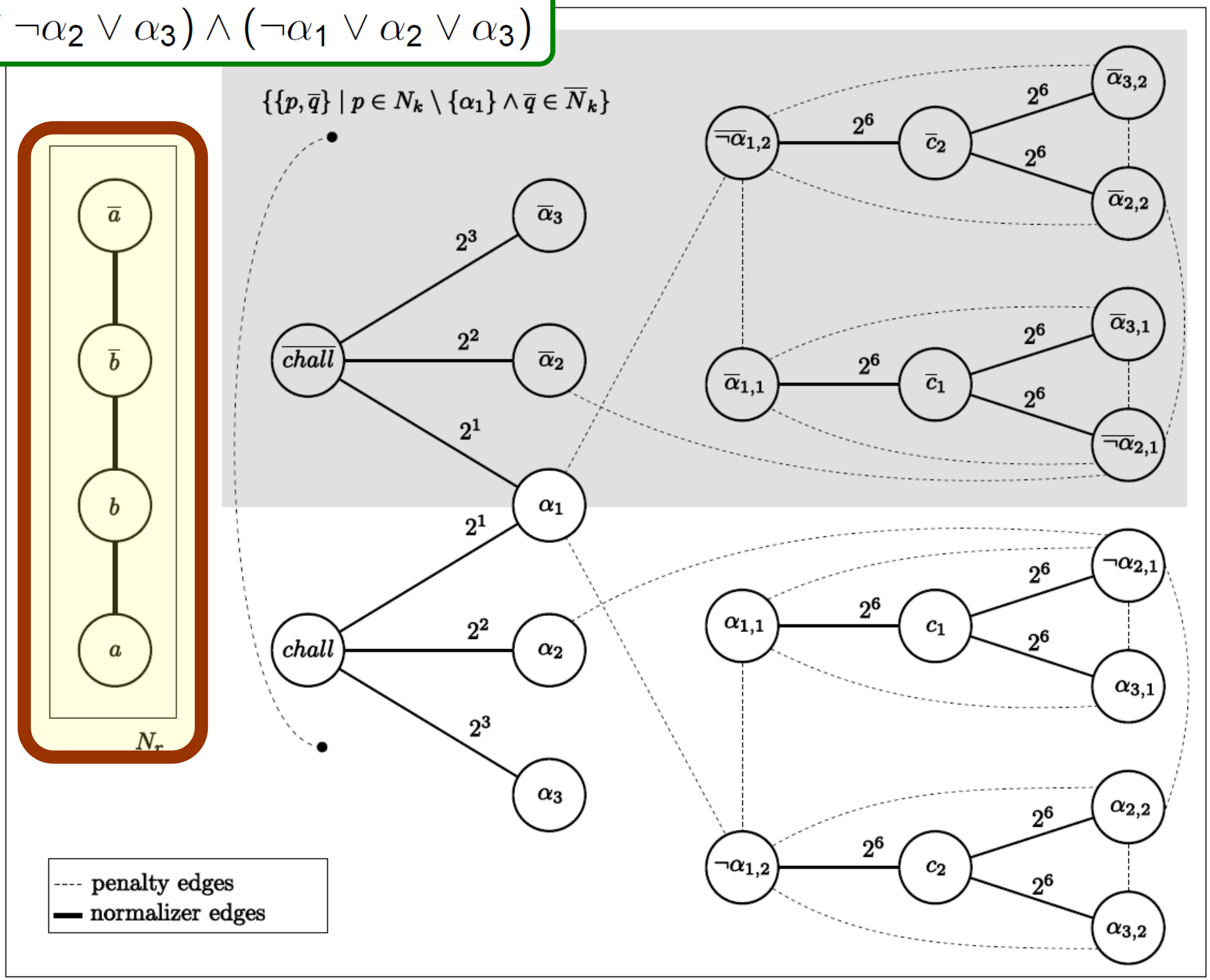
Overview of the Reduction

$$\hat{\phi} = (\alpha_1 \vee \neg\alpha_2 \vee \alpha_3) \wedge (\neg\alpha_1 \vee \alpha_2 \vee \alpha_3)$$



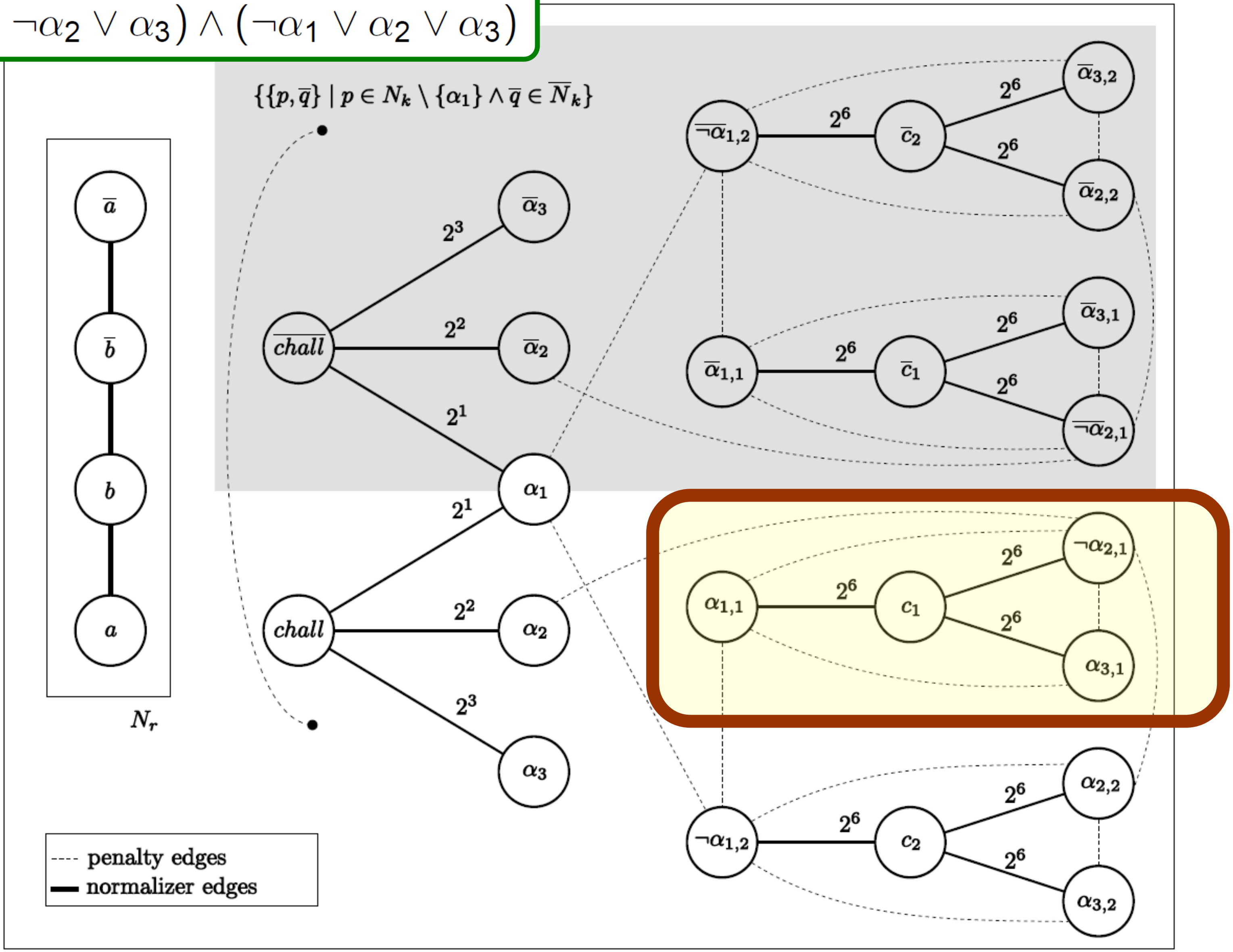
Overview of the Reduction

$$\hat{\phi} = (\alpha_1 \vee \neg\alpha_2 \vee \alpha_3) \wedge (\neg\alpha_1 \vee \alpha_2 \vee \alpha_3)$$



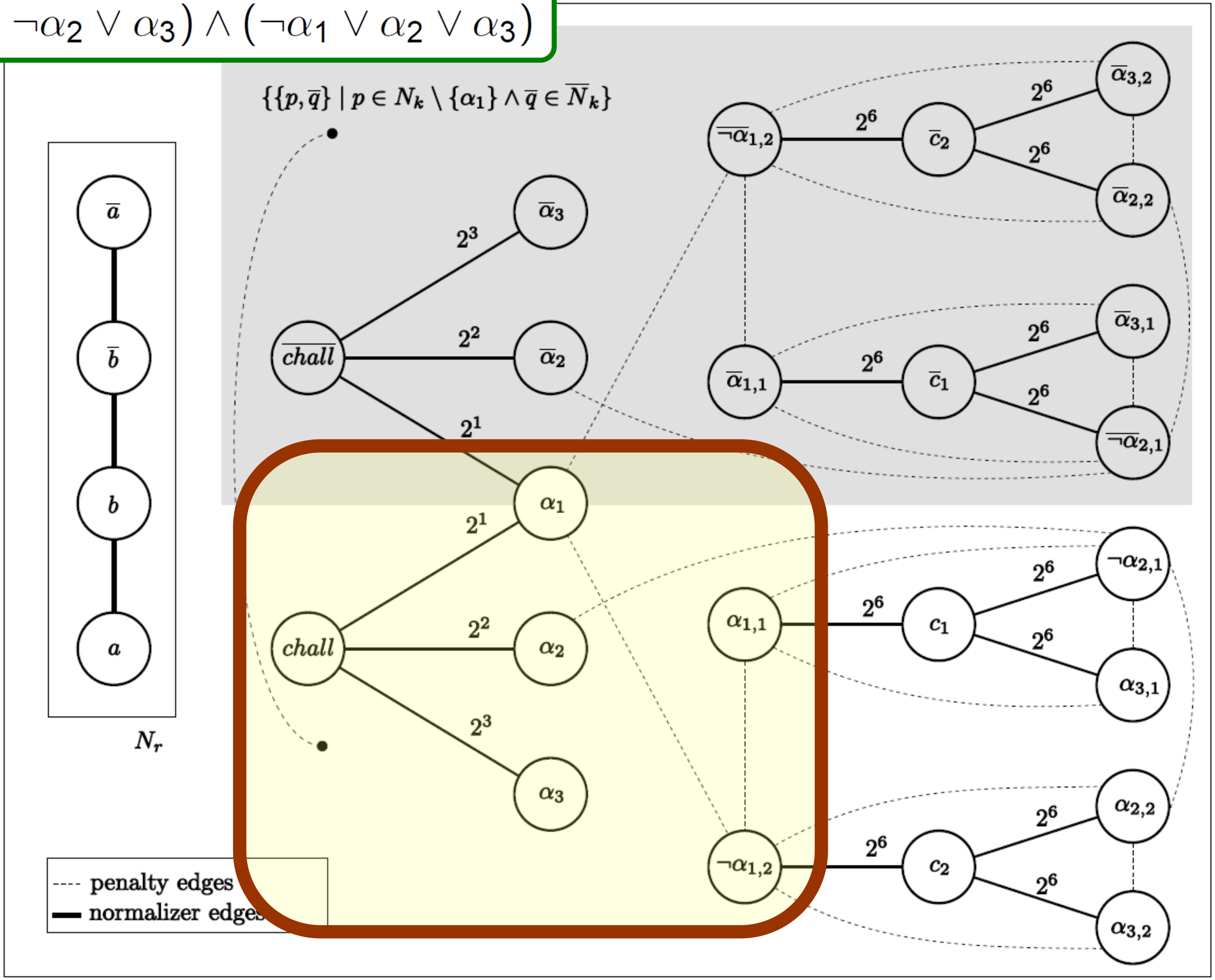
Overview of the Reduction

$$\hat{\phi} = (\alpha_1 \vee \neg\alpha_2 \vee \alpha_3) \wedge (\neg\alpha_1 \vee \alpha_2 \vee \alpha_3)$$



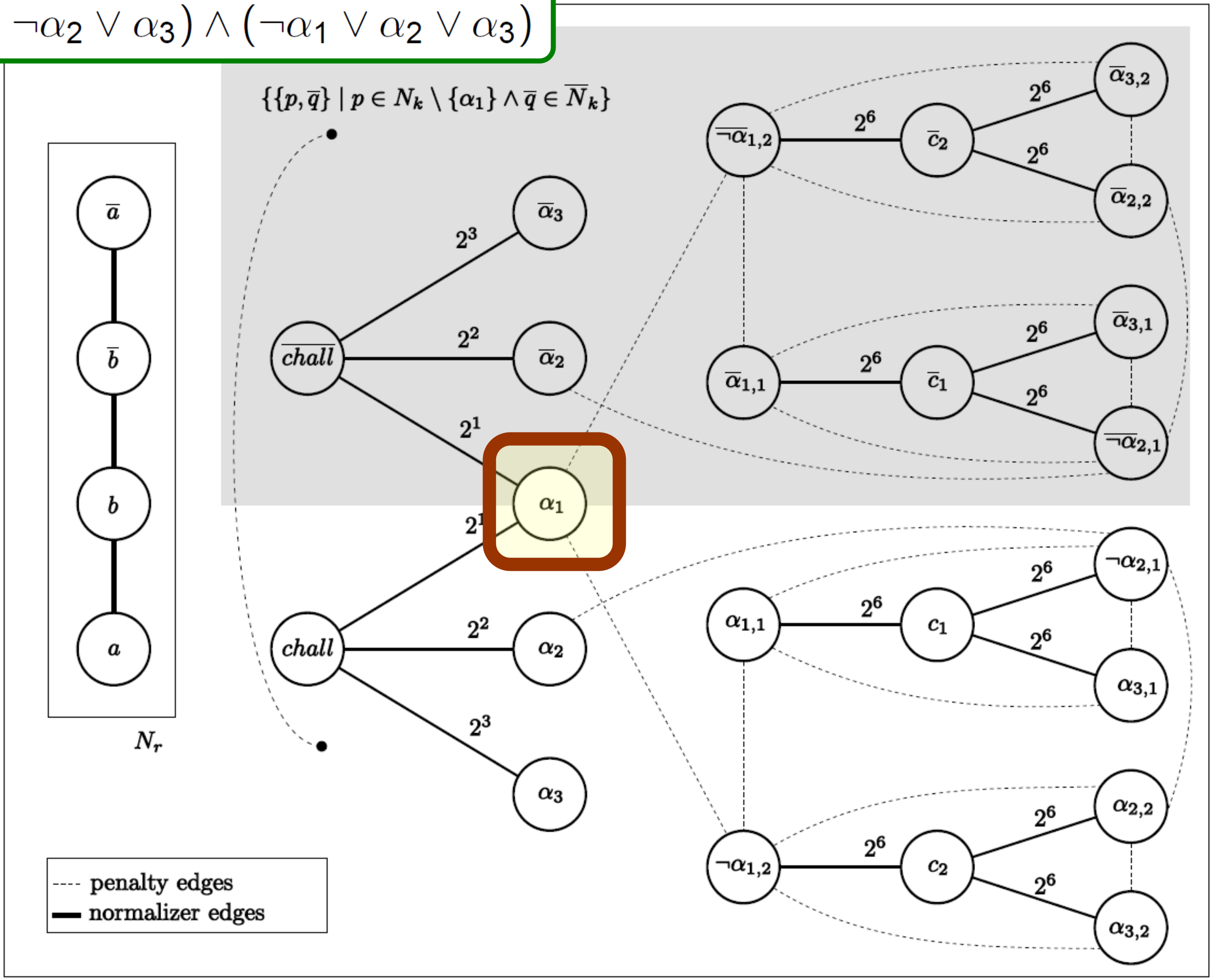
Overview of the Reduction

$$\hat{\phi} = (\alpha_1 \vee \neg\alpha_2 \vee \alpha_3) \wedge (\neg\alpha_1 \vee \alpha_2 \vee \alpha_3)$$



Overview of the Reduction

$$\hat{\phi} = (\alpha_1 \vee \neg\alpha_2 \vee \alpha_3) \wedge (\neg\alpha_1 \vee \alpha_2 \vee \alpha_3)$$



Thank you!