Conputational Foundations of Social Choice

A Complexity-Theoretic Perspective

My **sincere apologies** if you heard some of this already at my **COMSOC-08 tutorial**.

Jörg Rothe & Felix Brandt

HEINRICH HEINE UNIVERSITÄT DÜSSELDORF

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

ESF LogICCC Launch Conference, Prague, Czech Republic, 2008

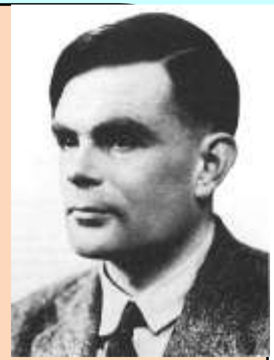
„Computer Science is not about computers,
any more than astronomy is about telescopes.“

Edsger Dijkstra

Outline

- **Impossibility Theorems in CS and SCT**
- **Complexity of Voting Problems:**
  - Winner Determination
  - Manipulation
- **Power-Index Comparison and Weighted Voting Games**
- **Multiagent Resource Allocation**

Impossibility Theorems in Computer Science and Social Choice Theory



<u>Computability Theory</u>

Alan Turing:

- Broke the Enigma-Code
- Invented the Turing machine
- Proved that some problems are **undecidable**

<u>Complexity Theory</u>

- Computational complexity of problems
- Lower bounds
- **Intractability** (NP-hardness)

<u>Social Choice Theory</u>

Ken Arrow: **No voting system** satisfying a certain small set of „fairness" conditions **can be nondictatorial.**

Gibbard-Satterthwaite Theorem: Manipulation is **unavoidable** in principle.

<u>Computational Social Choice</u>

- Computational complexity of social choice problems
- Manipulation can be **computationally hard**

Voting Problems:
How to Recruit a new Faculty Member

Candidates: A, B, C, D, E, F, G, H, I, J, K

Preferences of the Recruiting Committee:

J < A < B < E < D < F < G < H < K < I < C

I < J < A < D < E < F < G < B < C < K < H

A < B < F < G < H < K < I < C < J < E < D

E < G < F < B < J < I < H < C < A < D < K

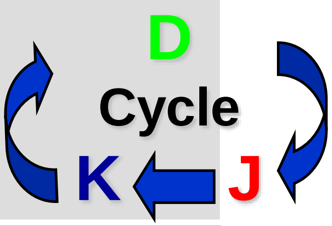
C < A < F < E < B < K < H < G < I < D < J

C < A < F < E < B < K < H < G < I < D < J

H < G < K < I < C < B < A < F < J < E < D

D < I < E < A < B < H < F < G < C < J < K

F < G < D < I < E < B < H < A < C < K < J



Make the List … by the Majority Rule:

Rank 1: D and J and K (aequo loco)

Since: D defeats J by 5:4 votes,
J defeats K by 5:4 votes,
K defeats D by 5:4 votes.

Condorcet's Paradox

Voting Problems:
Winner Determination, Manipulation, Control, Bribery

Winner Determination:

- How hard is it to determine the winners of a given election?
- For most election systems, it is easy to determine the winners, but for some it is hard (Carroll, Kemeny, and Young elections).

Manipulation:

- How hard is it, computationally, to manipulate the result of an election by strategic voting?
- The Gibbard-Satterthwaite Theorem says: Manipulation is unavoidable in principle.

Control:

- How hard is it, computationally, for an evil chair to influence the outcome of an election via procedural changes?

Bribery:

- How hard is it, computationally, for an external agent to bribe certain voters in order to change an election‘s outcome?

Voting Problems:

Winner Determination, Manipulation, Control, Bribery

Winner Determination: **Hardness is undesirable!**

- How hard is it to determine the winners of a given election?
- For most election systems, it is easy to determine the winners, but for some it is hard (Carroll, Kemeny, and Young elections).

Manipulation: **Hardness provides protection!**

- How hard is it, computationally, to manipulate the result of an election by strategic voting?
- The Gibbard-Satterthwaite Theorem says: Manipulation is unavoidable in principle.

Control: **Hardness provides protection!**

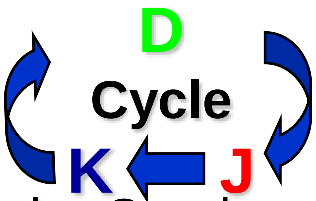
- How hard is it, computationally, for an evil chair to influence the outcome of an election via procedural changes?

Bribery: **Hardness provides protection!**

- How hard is it, computationally, for an external agent to bribe certain voters in order to change an election's outcome?

Winner Determination for Condorcet SCFs

- Majority Rule: A defeats B if A gets more votes than B.
- Condorcet Candidate: defeats each other candidate by majority.
- Condorcet Paradox:



- **Condorcet SCFs:** choose the Condorcet candidate (if one exists).

Lewis Carroll's Voting System (1876):

The winner is **whoever becomes a Condorcet candidate by a minimum number of sequential swaps of adjacent candidates.**

H. P. Young's Voting System (1977):

The winner is **whoever becomes a Condorcet candidate by removing a minimum number of voters.**

J. G. Kemeny's Voting System (1959):

The winner is **the candidate ranked first in a *consensus ranking***, **which minimizes the sum of the distances to the voters.**

Carroll Elections

- The **Carroll score** of a candidate **c** is the smallest number of sequential switches of adjacent candidates in the preference profile of the voters that make **c** a Condorcet candidate.
- **Carroll winner** is whoever has the lowest Carroll score.

Carroll Winner

Instance: A Carroll triple (***C,c,V***), where

C set of Candidates,

V preference profile of voters over ***C***,

c a designated candidate in ***C***.

Question: Is ***c*** a Carroll winner? That is, is it true that for all $d \in C$,

$$\mathrm{Score}(c) \leq \mathrm{Score}(d)\ ?$$

Carroll Score

Instance: A Carroll triple (***C,c,V***) and a positive integer ***k***.

Question: Is it true that $\mathrm{Score}(c) \leq k$?

Results for Carroll Election Problems

J. Bartholdi, C. Tovey & M. Trick (SCW 1989):

- Carroll Score **and** Kemeny Score **are NP-complete.**
- Carroll Winner and Kemeny Winner **are NP-hard.**

E. Hemaspaandra ... he (J.ACM 1997):

Carroll Winner is ... el access to NP."

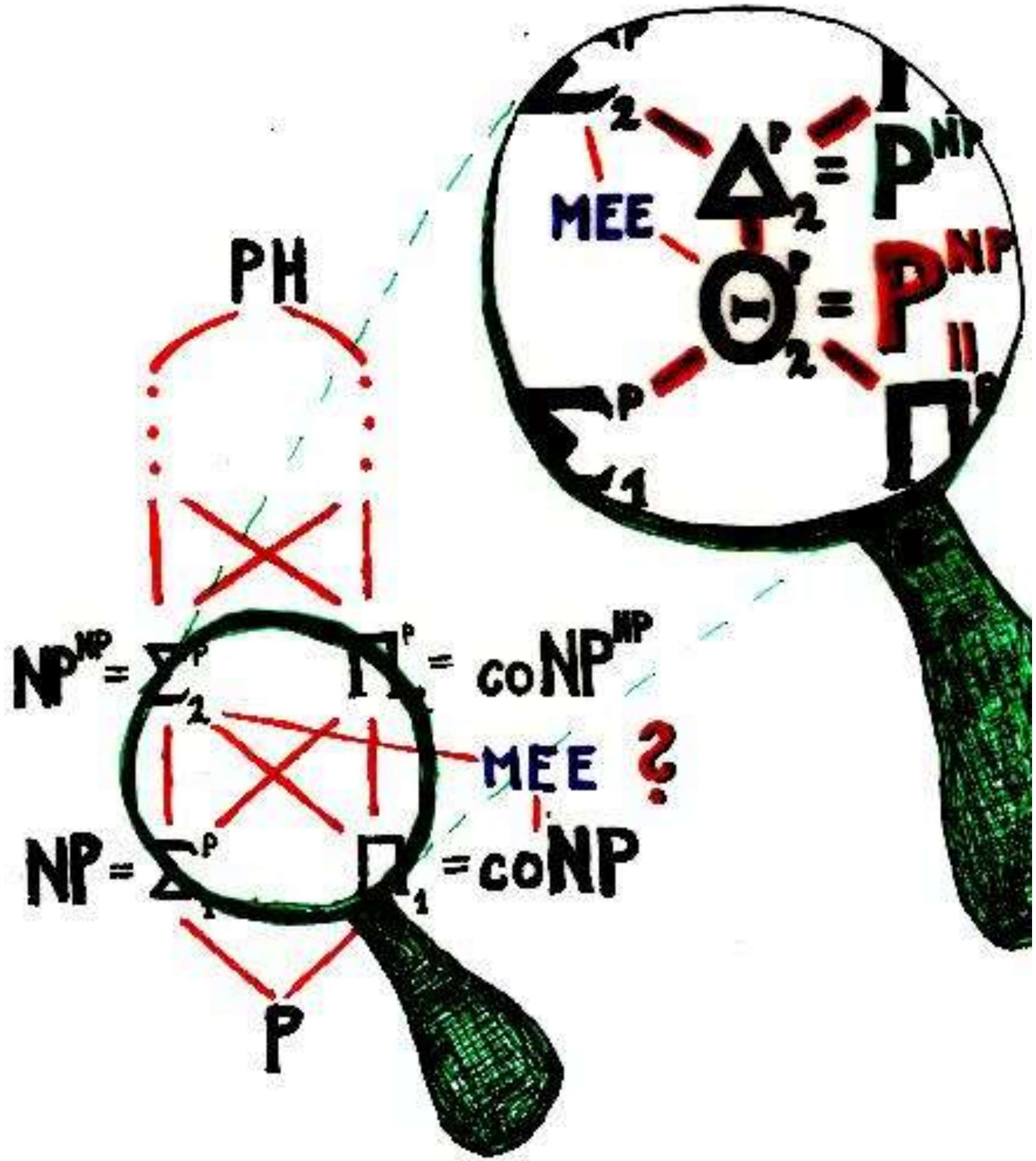
J. Rothe, ... CS 2003):

Yo ... e.

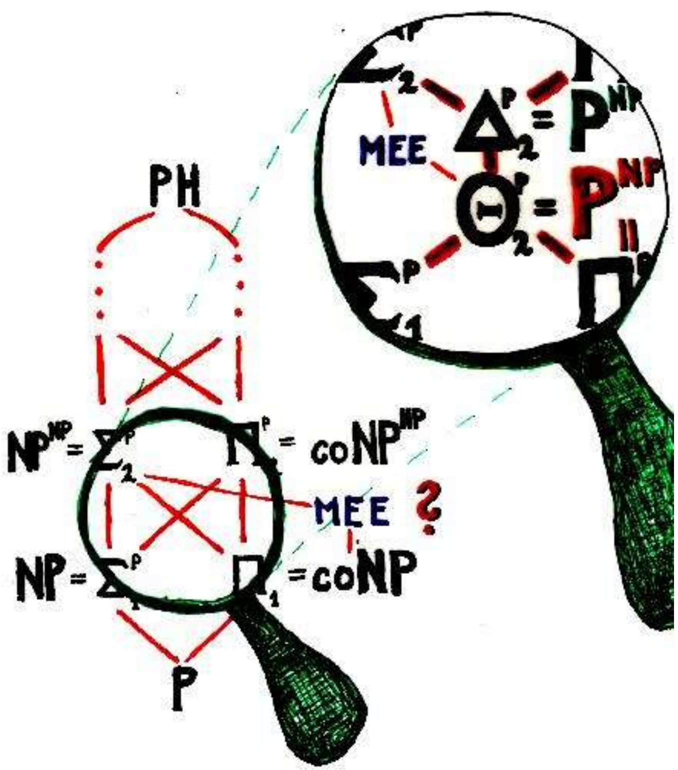
E. Hemaspaa ... el (TCS 2005):

Ke ... ete.

The Polynomial Hierarchy



Complexity of Solution Concepts/Choice Sets



F. Brandt & F. Fischer (MSS 2007):

Complexity of Minimal Covering Sets.

F. Brandt, F. Fischer & P. Harrenstein (TARK 2007):

Complexity of Choice Sets:

- Copeland Set
- Smith Set
- Schwartz Set
- von Neumann-Morgenstern stable sets
- Banks set
- Slater set

F. Brandt, F. Fischer, P. Harrenstein & M. Mair (AAAI 2008):

Computational Analysis of the Tournament Equilibrium Set.

Election Systems that are NP-hard to Manipulate

Gibbard-Satterthwaite: **Manipulation is unavoidable *in principle.***

Manipulation Problem

Instance: (***C,c,V***), where ***C*** is a set of candidates,
V is the voters' preference profile over ***C***,
c a designated candidate in ***C***.

Question: **Does there exist a preference order making *c* a winner?**

J. Bartholdi, C. Tovey & M. Trick (SCW 1989):

For Second-Order Copeland, **the winner problem is efficiently solvable, but the manipulation problem is NP-complete.**

V. Conitzer, T. Sandholm & J. Lang (J.ACM 2007):

- Studied *coalitional* manipulation by *weighted* voters
- Characterized the exact number of candidates for which manipulation becomes NP-hard for plurality, Borda, STV, Copeland, maximin, veto, and other protocols
- Considered both constructive and destructive manipulation

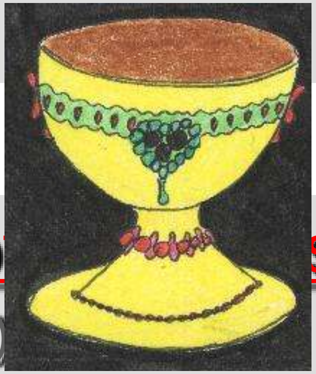
Election Systems that are NP-hard to Manipulate

E. Hemaspaandra & L. Hemaspaandra (JCSS 2007):

Provided the first dichotomy result for voting systems:

an easy-to-check condition („diversity of dislike“) that separates

- Scoring protocols that are NP-hard to manipulate from
- Scoring protocols that are easy to manipulate.



Holy Grail

Worst-Case vs. Average-Case/Frequency o… s

A. Procaccia & J. Rosenschein (JAIR 2007):

Junta distributions and Average-Case Complexity of Manipulation.

G. Erdélyi, L. Hemaspaandra, J. Rothe, H. Spakowski (FCT 2007):

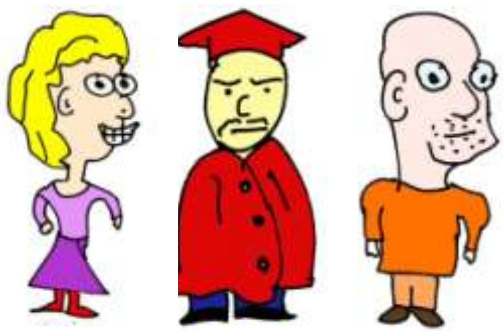
Frequency of Correctness vs. Average Polynomial Time.

M. Zuckerman, A. Procaccia & J. Rosenschein (SODA 2008):

Algorithms for the Coalitional Manipulation Problem.

Power-Index Comparison and Weighted Voting Games

Harvard University

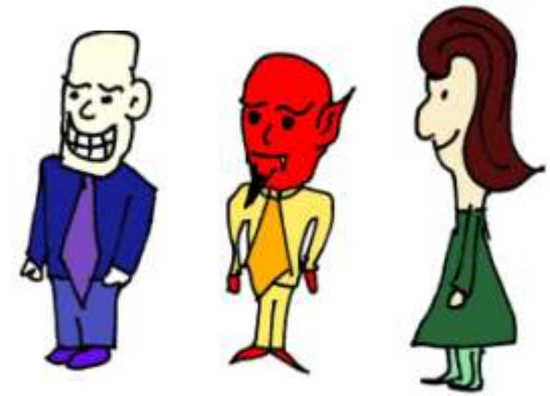


20 papers | 20 papers | 50 papers

Where will I have more (local) power?

4 papers
$10M

Money University

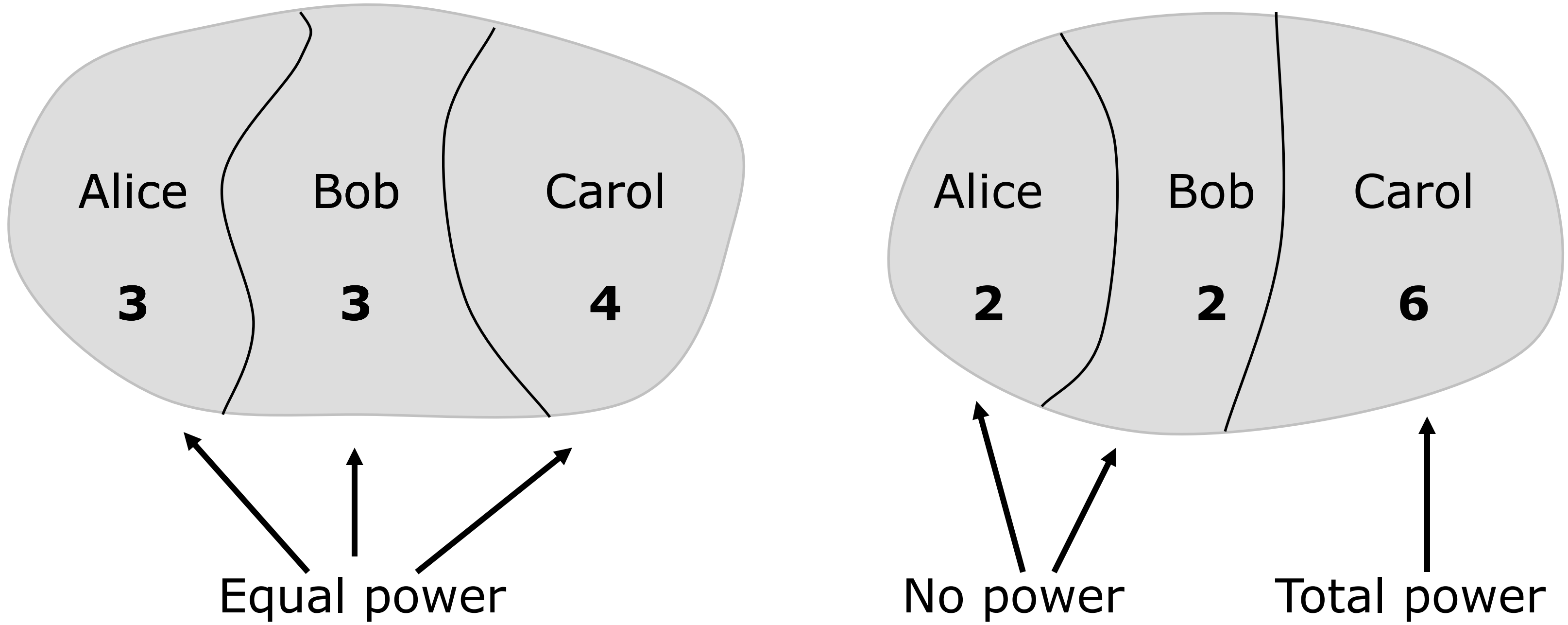


$2M | $5M | $2M

Aha! Clearly, I will have more (local) power at Money University!
But how else can I justify this choice?

Power-Index Comparison and Weighted Voting Games

Weighted Voting Games:



Power Index idea:

How "often" is the given player critical to the winning side?

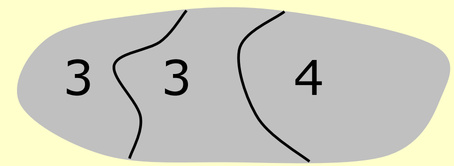
Power indices (e.g., **Shapley-Shubik** and **Banzhaf**) formally capture this idea. How hard is it to

- compute a power index for a given weighted voting game?
- compare the power index of two given weighted voting games?

Power Indices –
Banzhaf [1965] and Shapley-Shubik [1954]

Voting game: $G = (w_1, \ldots, w_n; q)$. Our notation:

- $N = \{1, \ldots, n\}$: set of players
- $w_1, \ldots, w_n$: weights of players
- q : quota value.



$q = 6$

Banzhaf*(G,i) = how many of the 2^{n-1} subsets of $N - \{i\}$ have total weight $< q$ but $\geq q - w_i$?

Banzhaf(G,i) = Banzhaf*(G,i)/2^{n-1}

(Probability that a randomly chosen coalition of players in $N - \{i\}$ is not successful but player i will put them over the top.)

SS*(G,i) = in how many of the n! permutations of N is i pivotal, i.e., the players before it sum to less than q but player i puts them over the top.

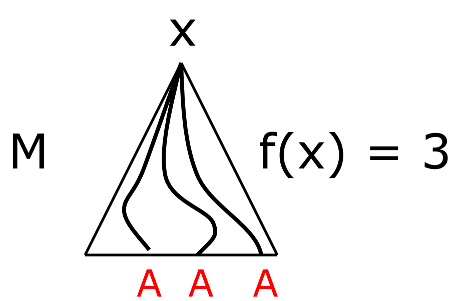
SS(G,i) = SS*(G,i)/n!

Complexity Classes: PP [Simon/Gill, 1970s] and #P [Valiant, 1979]

#P (Counting NP):

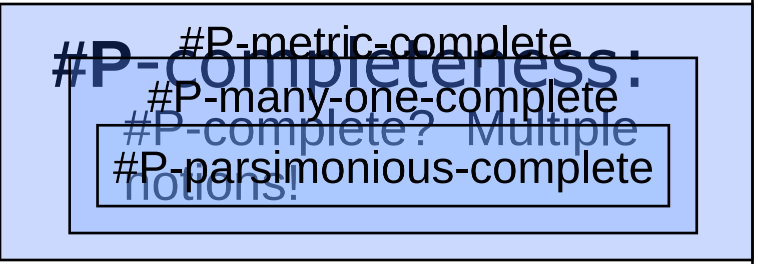
f ∈ **#P** if there is an NPTM such that

$(\forall x \in \Sigma^*)$[f(x) = number of accepting paths of M on input x].



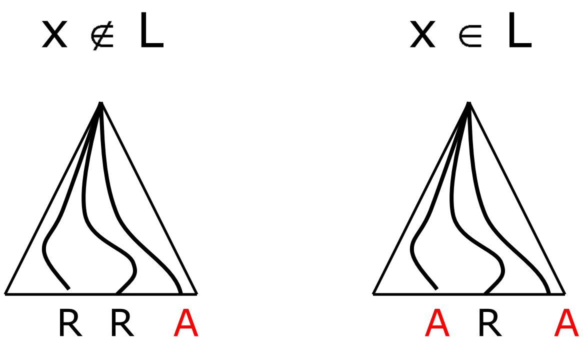
#P-completeness:

#P-complete? Multiple notions!



PP (Probabilistic Polynomial Time):

L ∈ **PP** if there is a probabilistic polynomial-time Turing machine that has acceptance probability greater than 50% precisely on the strings in L.



PP-completeness:

- Polynomial-Time Many-One Reducibility: $A \leq_m^P B \Leftrightarrow (\exists f \in FP)(\forall x \in \Sigma^*)[x \in A \Leftrightarrow f(x) \in B]$.
- B is hard for **PP** if $(\forall A \in \mathbf{PP})[A \leq_m^P B]$.
- B is **PP**-complete if B is in **PP** and is **PP**-hard.

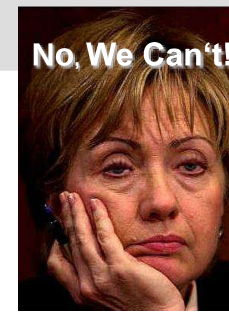
Results for Computing Power Indices

Prasad & Kelly (1990)+Hunt, Marathe, Radhakrishnan & Stearns (1998):

Banzhaf* is #P-parsimonious-complete.

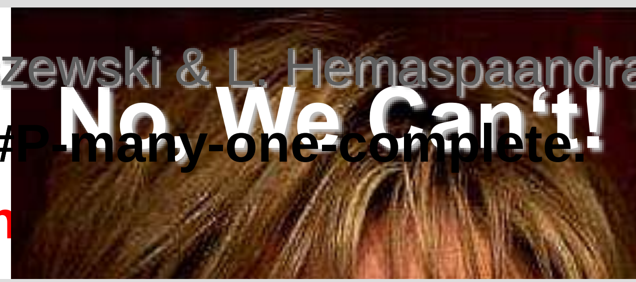
X. Deng & C. Papadimitriou (1994):

SS* is #P-metric-complete.



P. Faliszewski & L. Hemaspaandra (2008):

- **SS* is #P-many-one-complete.**
- **SS* is n … plete.**



PowerCompare-PI

(where **PI** is either **Banzhaf*** or **SS***)

Instance: Two weighted voting games, G and G′, and a player i.

Question: Is it true that **PI**(G, i) > **PI**(G′, i)?

(Can we improve this to #P-parsimonious-completeness?)

P. Faliszewski & L. Hemaspaandra (2008):

- **PowerCompare-Banzhaf* is PP-complete.**
- **PowerCompare-SS* is PP-complete.**

Multiagent Resource Allocation after World War II

- Set of Agents: the Allies of World War II
- Set of Resources: Germany‘s Federal States

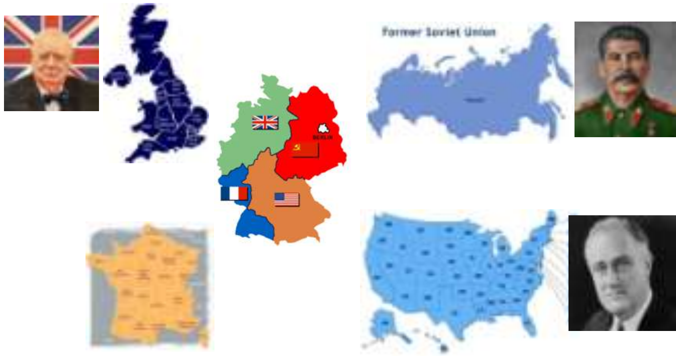
Multiagent Resource Allocation

- Set of Agents: $A = \{1, 2, ..., n\}$
- Set of Resources: $R = \{r_1, r_2, \ldots, r_m\}$
- Each agent *a* has
  - a preference $\leq_a$ over allocations
  - a utility function that assigns values to bundles of resources.
- Each resource is indivisible and nonsharable.
- An allocation is a mapping *P* from *A* to bundles of resources.
  Useful properties:
  - Envy-freeness
  - Pareto optimality

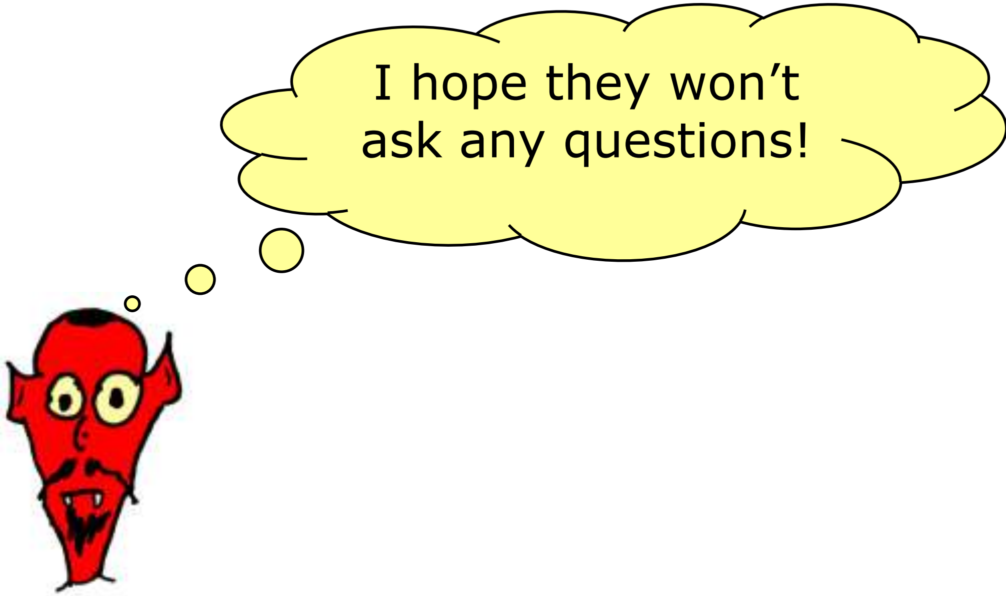
Chevaleyre, Dunne, Endriss, Lang, Lemaitre, Maudet, Padget, Phelps, Rodriguez-Aguilar & Sousa (2005):

MARA Survey.

•to determine if a given allocation is envy-free?



Thank you!



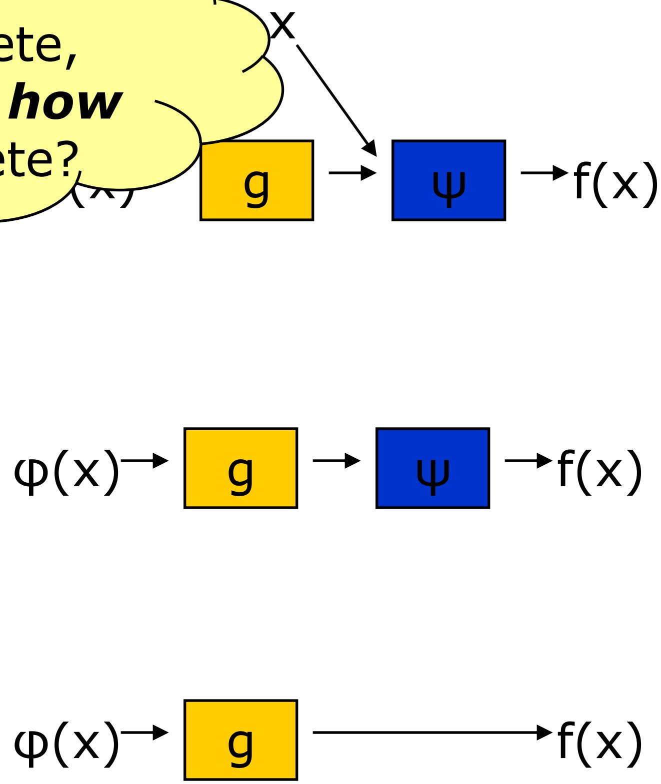
“Hardest” Problems for #P

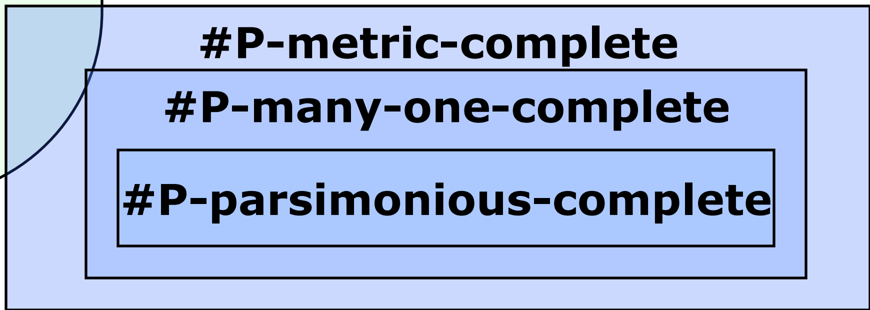
Definition:

1. [Krentel, 1988] A function $f:\Sigma^*\rightarrow N$
   reduces to a function $g:\Sigma^*\rightarrow N$ if th
   two FP functions, φ and ψ, such th
   $(\forall x\in\Sigma^*)[\ f(x) = \psi(\ x,\ g(\ \varphi(x)\ )\ )$

2. [Zankó, 1991] A function $f:\Sigma$ ... *-one reduces* to a function $g:\Sigma^*\rightarrow N$ ... ist
   two FP functions, φ and ψ, su
   $(\forall x\in\Sigma^*)[\ f(x) = \psi(\ g($ ... **.**

3. [Simon, 1975] A function $f:\Sigma^*\rightarrow N$ *parsimoniously reduces* to a function $g:\Sigma^*\rightarrow N$ if there exists an FP function φ such that
   $(\forall x\in\Sigma^*)[\ f(x) = g(\varphi(x))\ ].$





Multiagent Resource Allocation

- Set of Agents: $A = \{1, 2, ..., n\}$
- Set of Resources: $R = \{r_1, r_2, \ldots, r_m\}$
- Each agent *a* has
  - a preference $\leq_a$ over allocations
  - a utility function that assigns values to bundles of resources.
- Each resource is indivisible and nonsharable.
- An allocation is a mapping *P* from *A* to bundles of resources.
  Useful properties:
  - **Envy-freeness**
  - **Pareto optimality**
- An allocation P is envy-free if every agent is at least as happy with its share as with any of the other agents' shares.

Formally: $$(\forall a,b \in A)\left[P(b) \leq_a P(a)\right]$$

- An allocation P is Pareto optimal if it is not Pareto-dominated by any other allocation. That is, for no allocation Q does it hold that

$$(\forall a \in A)\left[P \leq_a Q\right] \wedge (\exists b \in A)\left[P <_b Q\right]$$