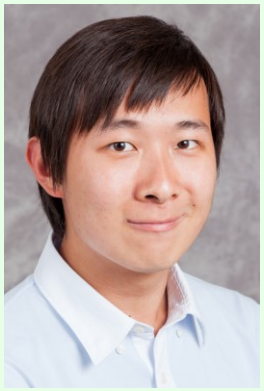
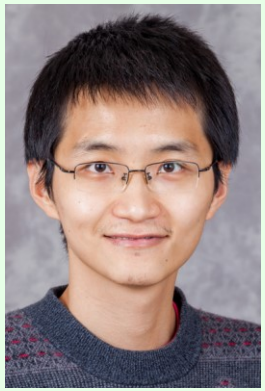
CPS 590.2: Computational Microeconomics: Game Theory, Social Choice, and Mechanism Design

https://www2.cs.duke.edu/courses/fall18/compsci590.2/

Instructor: Vincent Conitzer
(Kimberly J. Jenkins University Professor of New Technologies
Professor of Computer Science, Professor of Economics, and Professor of Philosophy)
conitzer@cs.duke.edu

TAs: Harsh Parikh and Hanrui Zhang; also helping: postdoc Yu Cheng





Journal, conference, …

ACM Transactions on Economics and Computation (TEAC)

History

John von Neumann

computer architecture (von Neumann architecture)

Computer Science & Engineering

game theory (minimax theorem)

Economic Theory

linear programming (duality)

Mathematical Optimization & Operations Research

?

1900 *1950* *2000*

CS-ECON

http://econ.cs.duke.edu

CS-ECON@DUKE
Exploring the Intersection of Computer Science and Economics

Home | Schedule | Past Talks | People | Mini-retreat | Reading Group

Who Are We?

We are a group of Duke University faculty, postdocs, and students interested in the intersection of computer science and economics (and the social sciences more broadly) and the impact of this interplay on decisions in information technology and digital business. This includes applying techniques from computer science and optimization to economics -- for example, using computation to design market clearing mechanisms and to implement efficient allocation and pricing in them -- as well as applying techniques from economics to computer science -- for example, designing incentives for users of networked computer systems and social networks.

Contacts

For organizational questions about the seminar series:

- Yuqian Li
- Catherine Moon

For other matters, contact the relevant faculty member(s):

- Atila Abdulkadiroglu (Econ)
- Vincent Conitzer (CS)
- Rachel Kranton (Econ)
- Ben Lee (ECE)
- Kamesh Munagala (CS)

CS-Econ Talks

- Upcoming Talks
- Past Talks

Related Seminars

- AI Group (CS)
- Algorithms Seminar (CS)
- Decision Sciences Seminar (Fuqua)
- Duke Robotics, Intelligence, and Vision (DRIV) Seminar (CS)
- Machine Learning
- Microeconomic Theory Seminars (Econ)

For Prospective Students
Degree Programs
M.A. Economics
M.A. Analytical Political Economy
M.S. Economics & Computation
M.S. Quantitative Financial Economics
For Current Students
EcoTeach: Student Services Center

M.S. Economics & Computation

The joint field of economics and computer science has emerged from two converging intellectual needs: Computer science has become increasingly important for economists working with big data to address complex questions. Students interested in learning about computational mechanism design with applications to economics are ideal candidates for this program. Students whose interest is more generally focused on data analytics across a broad range of fields may also be interested in Duke's Master of Quantitative Management (MQM) program, offered at the Fuqua School of Business, and/or Duke's new Master in Interdisciplinary Data Science (MIDS) program, which is accepting its first class in Fall 2018.

The MSEC program combines the strengths of the Departments of Economics and Computer Science to educate students in these important computational skills linked to economics, and to prepare them for Ph.D. studies or careers in economics, finance, government, and business. Reflecting this strong interdisciplinary relationship, Duke University ranks No. 5 for research in economics and computation, according to CSRankings.org.

This program is designed to meet the needs of students with varied levels of exposure to either field, but a strong quantitative background is recommended.

What is Economics?

- "Economics is the social science that studies the production, distribution, and consumption of goods and services." [Wikipedia, Aug. 2018]
- Some key concepts:
  - Economic agents or players (individuals, households, firms, …)
  - Agents' current endowments of goods, money, skills, …
  - Possible outcomes ((re)allocations of resources, tasks, …)
  - Agents' preferences or utility functions over outcomes
  - Agents' beliefs (over other agents' utility functions, endowments, production possibilities, …)
  - Agents' possible decisions/actions
  - Mechanism that maps decisions/actions to outcomes

An economic picture

v( ) = 200

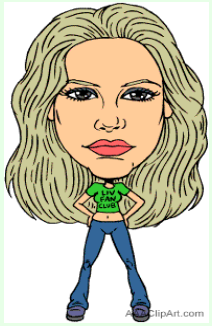
$ 800

v( ) = 100

v( ) = 400

v( ) = 200

v( ) = 400



$ 600

$ 200

After trade (a more efficient outcome)

v( ) = 200

$ 1100

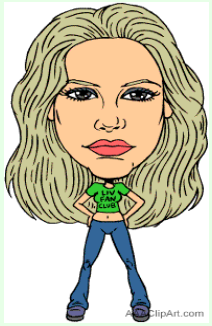
… but how do we get here? Auctions? Exchanges? Unstructured trade?

v( ) = 100

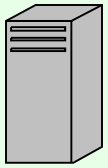
v( ) = 400

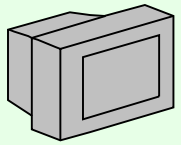
v( ) = 200

v( ) = 400



$ 400





$ 100

Some distinctions in economics

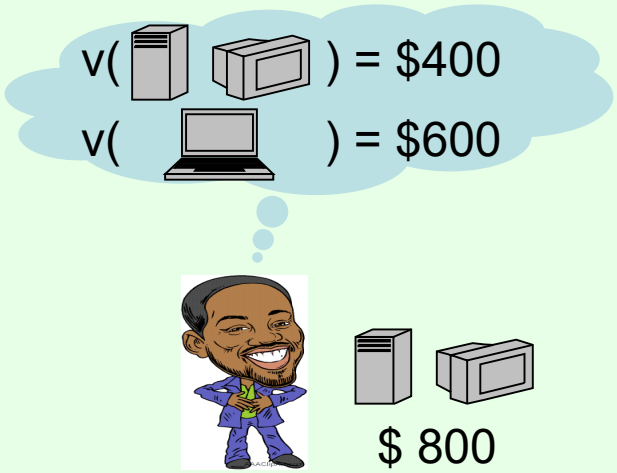
- Descriptive vs. normative economics
  - Descriptive:
    - seeks only to describe real-world economic phenomena
    - does not care if this is in any sense the "right" outcome
  - Normative:
    - studies how people "should" behave, what the "right" or "best" outcome is
- Microeconomics vs. macroeconomics
  - Microeconomics: analyzes decisions at the level of individual agents
    - deciding which goods to produce/consume, setting prices, …
    - "bottom-up" approach
  - Macroeconomics: analyzes "the sum" of economic activity
    - interest rates, inflation, growth, unemployment, government spending, taxation, …
    - "big picture"

What is Computer Science?

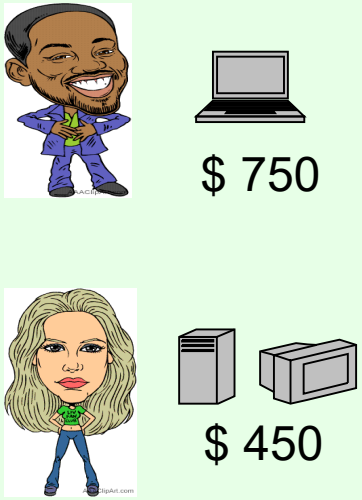
- "Computer science is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers. […] A computer scientist specializes in the theory of computation and the design of computational systems." [Wikipedia, Aug. 2018]
- A computational problem is given by a function f mapping inputs to outputs
  - For integer x, let f(x) = 0 if x is prime, 1 otherwise
  - For an initial allocation of resources x, let f(x) be the (re)allocation that maximizes the sum of utilities
- An algorithm is a fully specified procedure for computing f
  - E.g., sieve of Eratosthenes
  - A correct algorithm always returns the right answer
  - An efficient algorithm returns the answer fast
- Computer science is also concerned with building larger artifacts out of these building blocks (e.g., personal computers, spreadsheets, the Internet, the Web, search engines, artificial intelligence, …)

Resource allocation as a computational problem

input



output



Here, gains from trade ($300) are divided evenly (not essential)

Economic mechanisms

"true" input

v( ) = $400

v( ) = $600

$ 800

agent 1's bidding algorithm

agents' bids

v( ) = $500

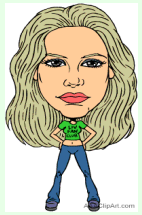
v( ) = $501

$ 800

result

$ 800

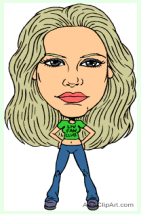
exchange mechanism (algorithm)



$ 400

v( ) = $500

v( ) = $400

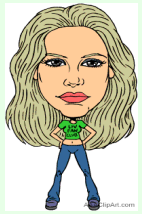


$ 400

agent 2's bidding algorithm

v( ) = $451

v( ) = $450



$ 400

Exchange mechanism designer does not have direct access to agents' private information

Agents will selfishly respond to incentives

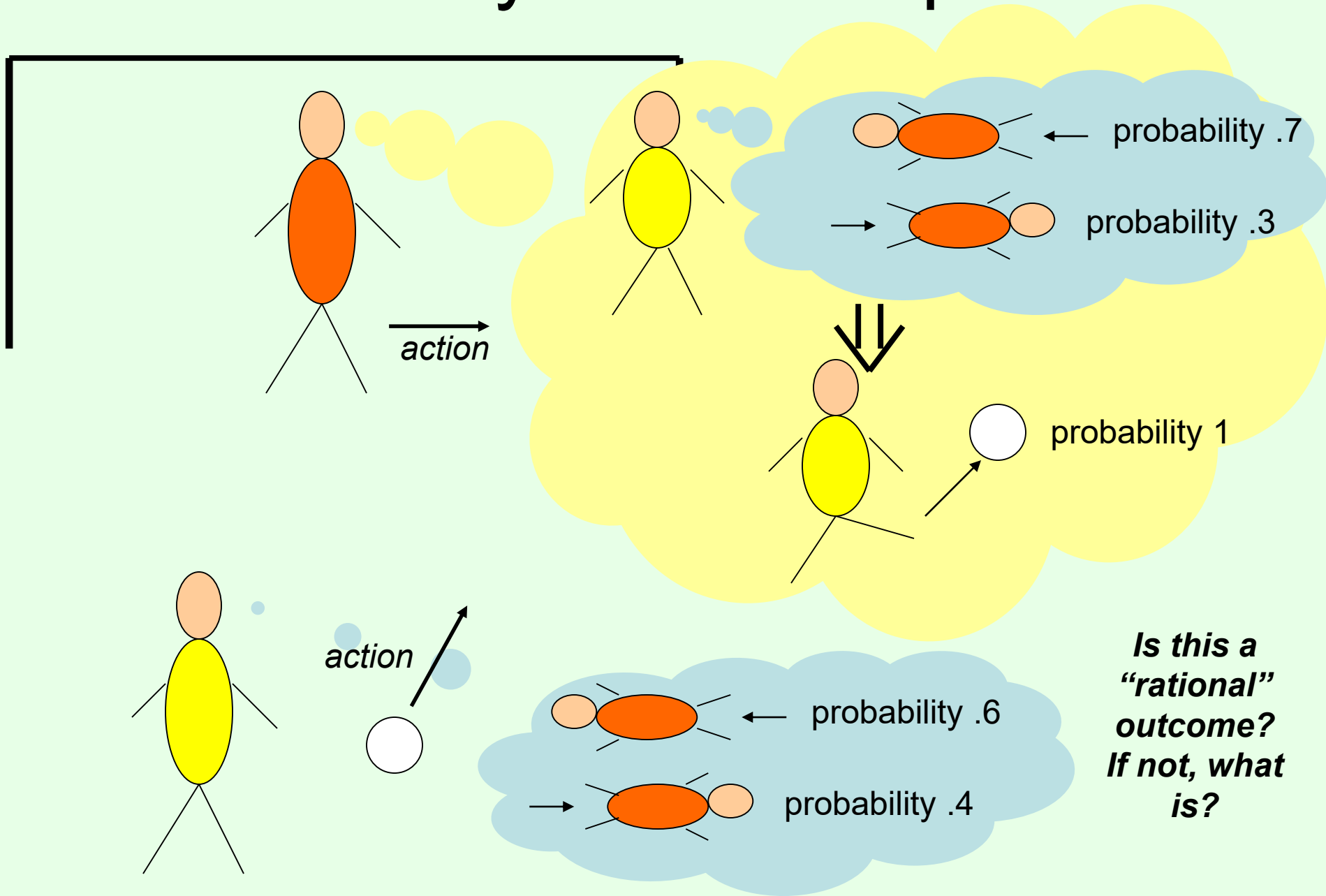
What is game theory?

- “Game theory is the study of mathematical models of strategic interaction between rational decision-makers. It has applications in all fields of social science, as well as in logic and computer science. […] Today, game theory applies to a wide range of behavioral relations, and is now an umbrella term for the science of logical decision making in humans, animals, and computers.” [Wikipedia, Aug. 2018]

What is game theory…

- Game theory studies settings where multiple parties (agents) each have
  - different preferences (utility functions),
  - different actions that they can take
- Each agent's utility (potentially) depends on all agents' actions
  - What is optimal for one agent depends on what other agents do
    - Very circular!
- Game theory studies how agents can rationally form beliefs over what other agents will do, and (hence) how agents should act
  - Useful for acting as well as predicting behavior of others

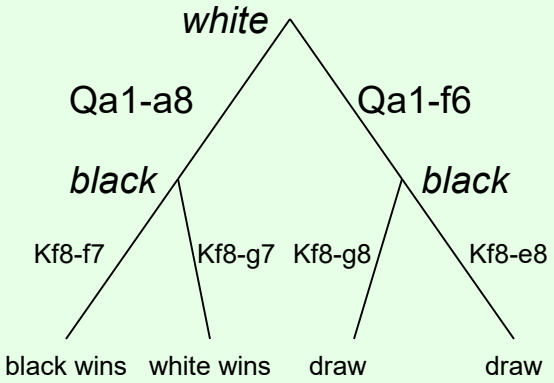
Penalty kick example



Is this a "rational" outcome? If not, what is?

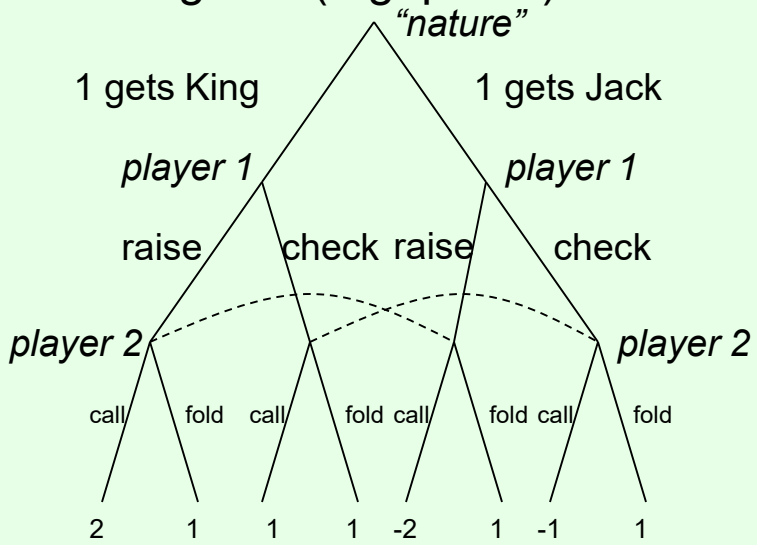
Game playing & AI

perfect information games: no uncertainty about the state of the game (e.g. tic-tac-toe, chess, Go)



- Optimal play: value of each node = value of optimal child for current player (backward induction, minimax)
- For chess and Go, tree is too large
  - Use other techniques (heuristics, limited-depth search, alpha-beta, deep learning, …)
- Top computer programs better than humans in chess, ~~not yet in Go~~

imperfect information games: uncertainty about the state of the game (e.g. poker)



- Player 2 cannot distinguish nodes connected by dotted lines
  - Backward induction fails; need more sophisticated game-theoretic techniques for optimal play
- Small poker variants can be solved optimally
- Humans still better than top computer programs at full-scale poker (at least most versions)?
- Top computer (heads-up) poker players are based on techniques for game theory

Real-world security applications

Milind Tambe's TEAMCORE group (USC)

Airport security

Where should checkpoints, canine units, etc. be deployed?

Federal Air Marshals

Which flights get a FAM?

US Coast Guard

Which patrol routes should be followed?

Wildlife Protection

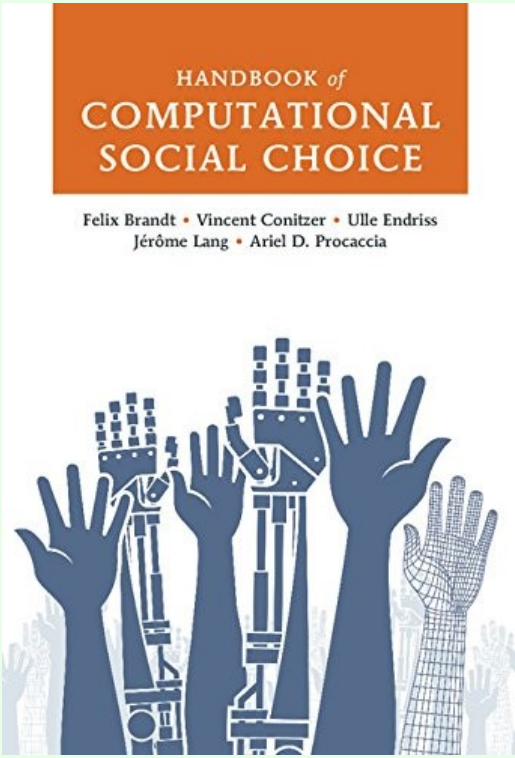
Where to patrol to catch poachers or find their snares?

Questions and problems in (computational) game theory

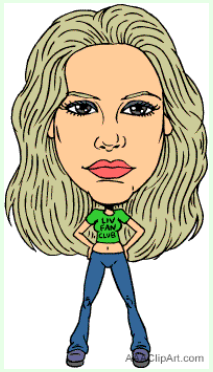
- How should we represent games (=strategic settings)?
  - Standard game-theoretic representations not always concise enough
- What does it mean to solve a game?
  - Solution concepts from game theory, e.g., Nash equilibrium
- How computationally hard is it to solve games?
  - Can we solve them approximately?
- Is there a role for (machine) learning in games?
- What types of modeling problems do we face when addressing real-world games?
  - E.g., applications in security
- …

What is social choice?

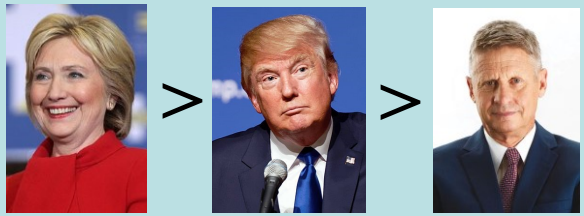
- "Social choice theory or social choice is a theoretical framework for analysis of combining individual opinions, preferences, interests, or welfares to reach a *collective decision* or *social welfare* in some sense." [Wikipedia, Aug. 2018]
- I.e., making decisions based on the preferences of multiple agents
- Largely, but not exclusively, focused on voting



Voting over outcomes

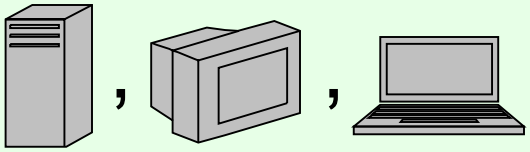


voting rule (mechanism) determines winner based on votes

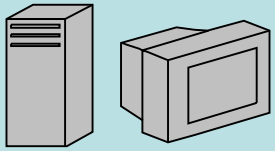


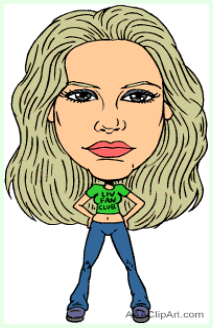
- Can vote over other things too
  - Where to go for dinner tonight, other joint plans, …
- Many different rules exist for selecting the winner

Combinatorial auctions

Simultaneously for sale:  , ,

bid 1

$v($  $) = \$500$



bid 2

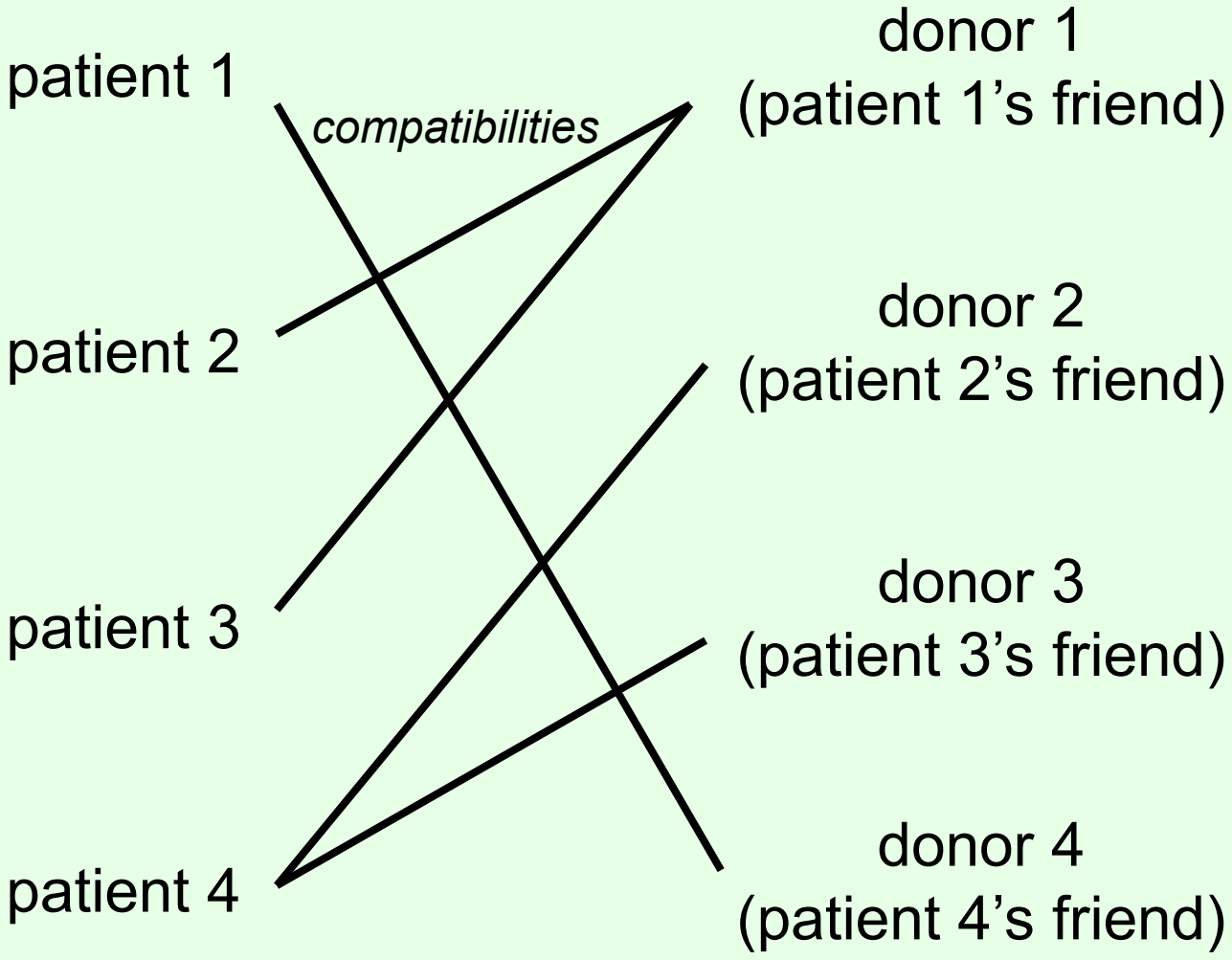
$v($  $) = \$700$

bid 3

$v($  $) = \$300$

used in truckload transportation, industrial procurement, radio spectrum allocation, …

Kidney exchange



Problems in computational social choice

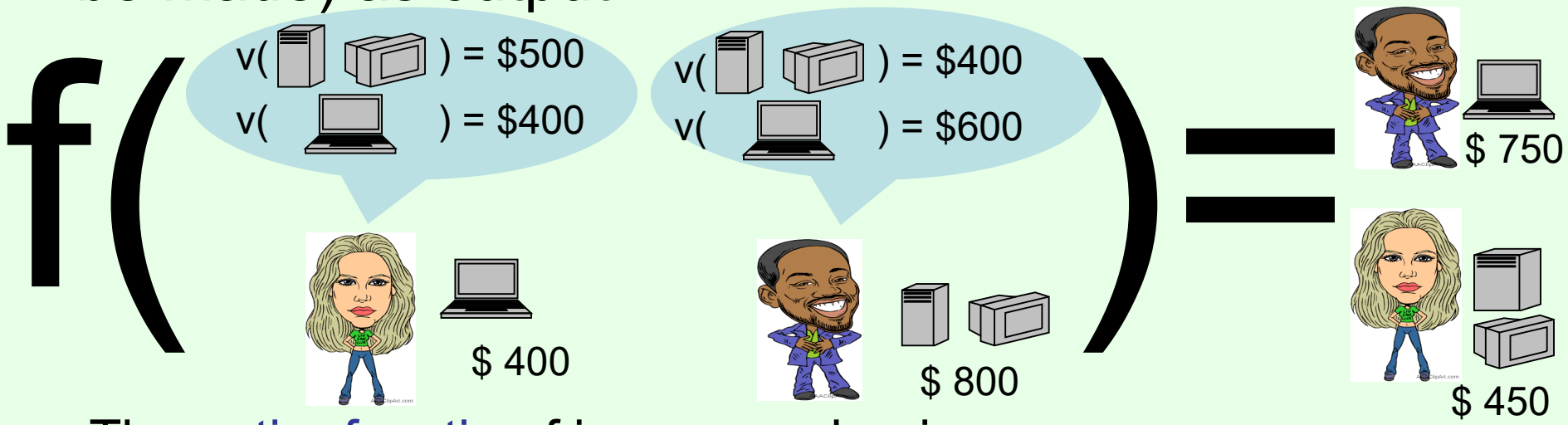
- Winner determination problem
  - For some voting rules, determining the winner is NP-hard
  - In a combinatorial auction, deciding which bids win is (in general) an NP-hard problem
- Preference elicitation (communication) problem
  - Can be impractical to communicate all of one's preferences (e.g., valuation for every bundle)
- Mechanism design problem
  - How do we get the bidders to behave so that we get good outcomes?
- These problems interact in nontrivial ways
  - E.g. limited computational or communication capacity can limit mechanism design options
  - … but can perhaps also be used in a positive way

What is mechanism design?

- "Mechanism design is a field in economics and game theory that takes an engineering approach to designing economic mechanisms or incentives, toward desired objectives, in strategic settings, where players act rationally. […] Two distinguishing features of [mechanism design] are:
  - that a game "designer" chooses the game structure rather than inheriting one
  - that the designer is interested in the game's outcome
- [Wikipedia, Aug. 2018]

Mechanism design…

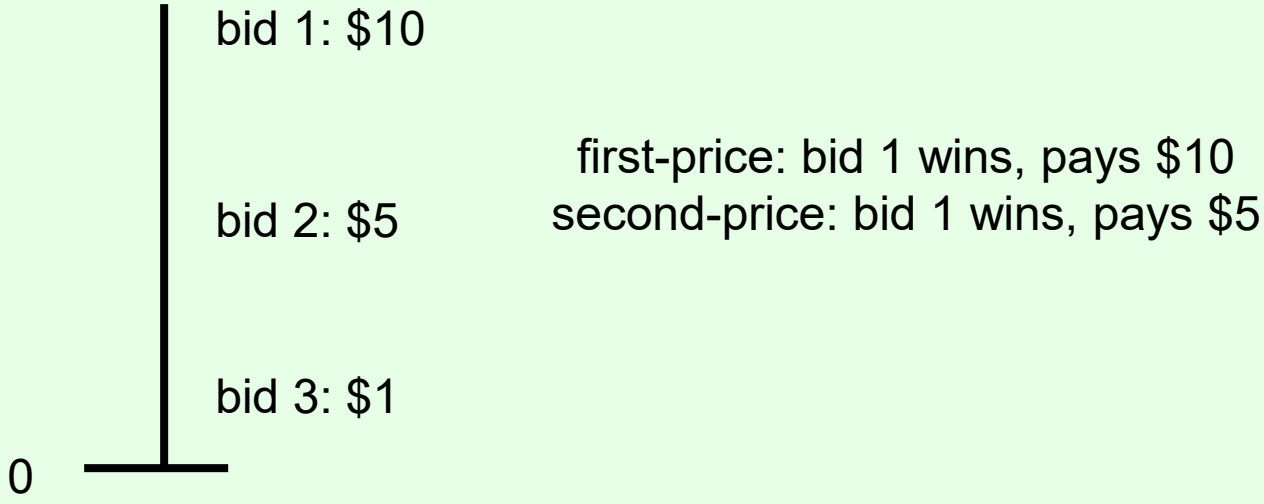
- Mechanism = rules of auction, exchange, …
- A function that takes reported preferences (bids) as input, and produces outcome (allocation, payments to be made) as output



- The entire function f is one mechanism
- E.g., the mechanism from before: find allocation that maximizes (reported) utilities, distribute (reported) gains evenly
- Other mechanisms choose different allocations, payments

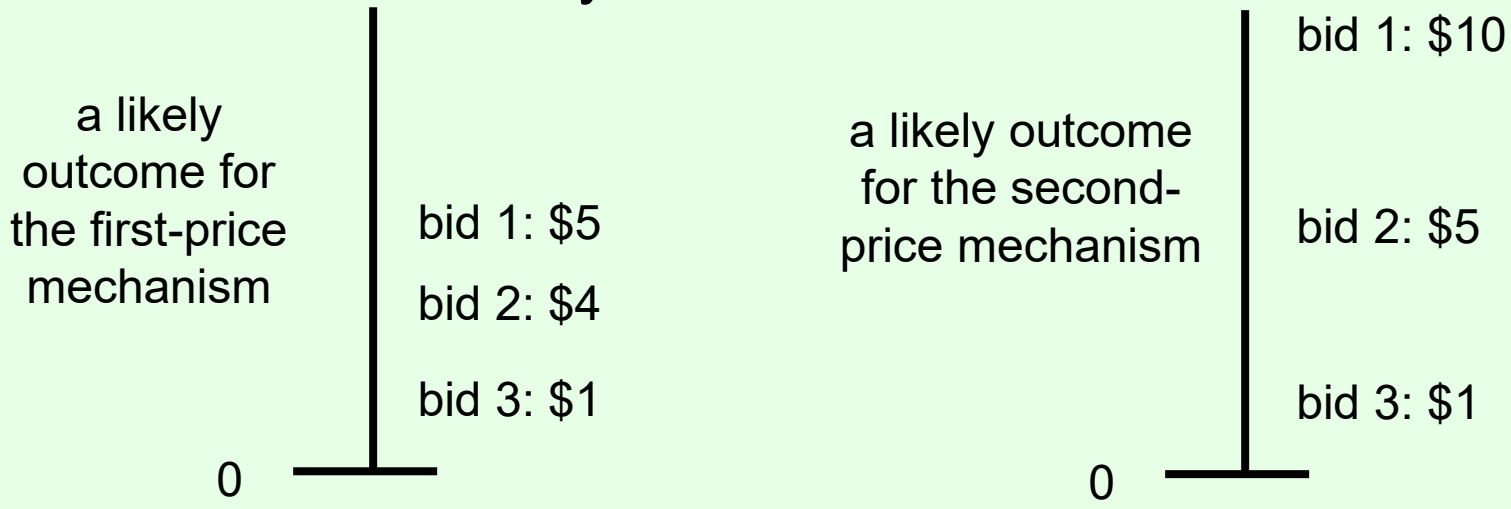
Example: (single-item) auctions

- Sealed-bid auction: every bidder submits bid in a sealed envelope
- First-price sealed-bid auction: highest bid wins, pays amount of own bid
- Second-price sealed-bid auction: highest bid wins, pays amount of second-highest bid



Which auction generates more revenue?

- Each bid depends on
  - bidder's true valuation for the item (utility = valuation - payment),
  - bidder's beliefs over what others will bid (→ game theory),
  - and... the auction mechanism used
- In a first-price auction, it does not make sense to bid your true valuation
  - Even if you win, your utility will be 0…
- In a second-price auction, (we will see later that) it always makes sense to bid your true valuation



Are there other auctions that perform better? How do we know when we have found the best one?

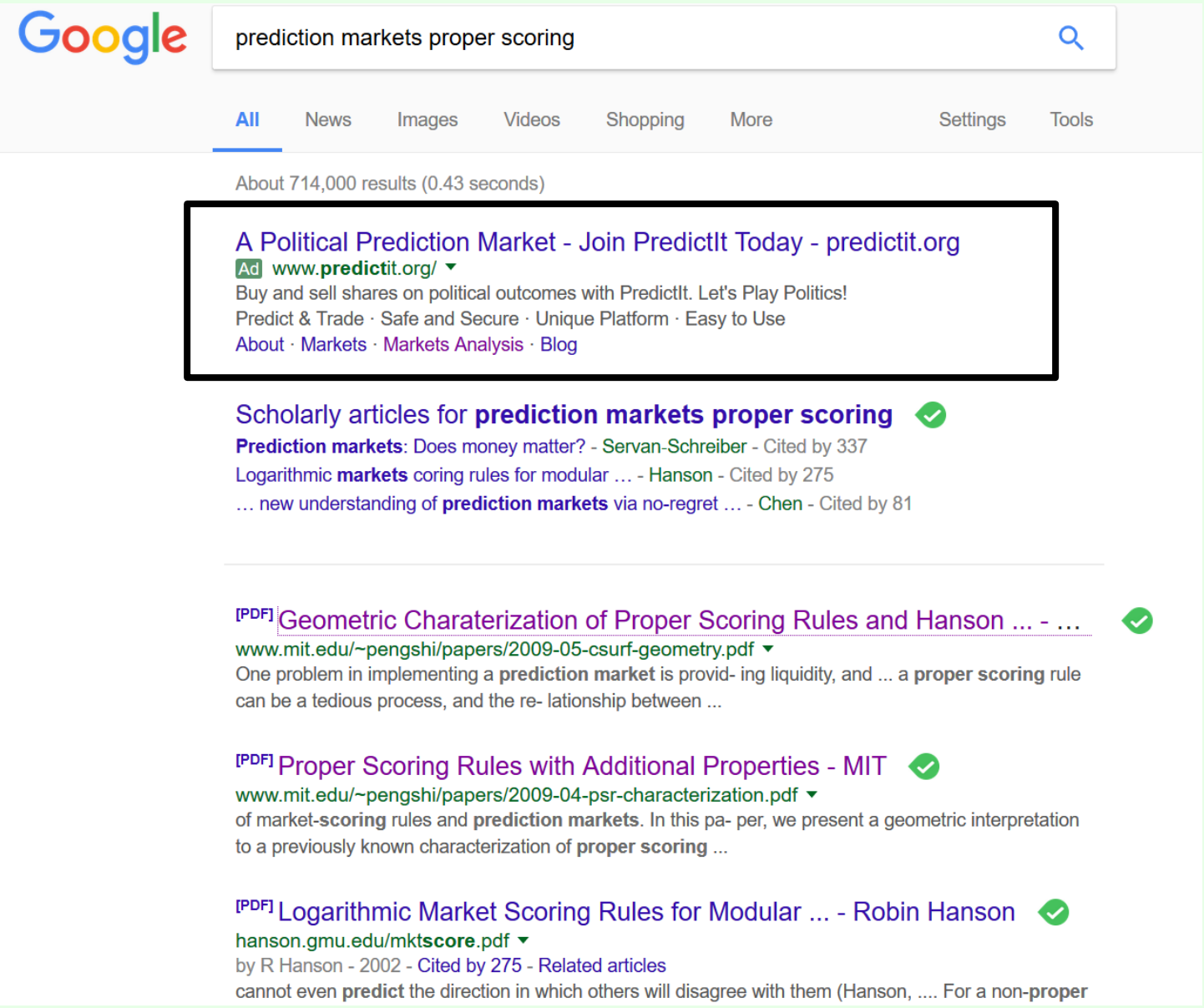
Mechanism design…

- Mechanism = game
- → we can use game theory to predict what will happen under a mechanism
  - if agents act strategically
- When is a mechanism "good"?
  - Should it result in outcomes that are good for the reported preferences, or for the true preferences?
  - Should agents ever end up lying about their preferences (in the game-theoretic solution)?
  - Should it always generate the best allocation?
  - Should agents ever burn money?(!?)
- Can we solve for the optimal mechanism?

Many uses of linear programming, mixed integer (linear) programming in this area

	Linear programming	**Mixed integer linear programming**
Game theory	Dominated strategies Minimax strategies Correlated equilibrium Optimal mixed strategies to commit to	Nash equilibrium Optimal mixed strategies to commit to in more complex settings
Social choice, expressive marketplaces	Winner determination in auctions, exchanges, … with partially acceptable bids	Winner determination in: auctions, exchanges, … without partially acceptable bids; Kemeny, Slater, other voting rules; kidney exchange
Mechanism design	Automatically designing optimal mechanisms that use randomization	Automatically designing optimal mechanisms that do not use randomization

Sponsored search / ad auctions



- Choice of ads (if any) to show determined by:
  - Advertiser bid
  - Predicted likelihood of click

Learning & mechanism design

- Auctioneer may need to know distribution over private information to design optimal mechanism
  - Learn from repeated play?
  - Exploration/exploitation tradeoffs?

1

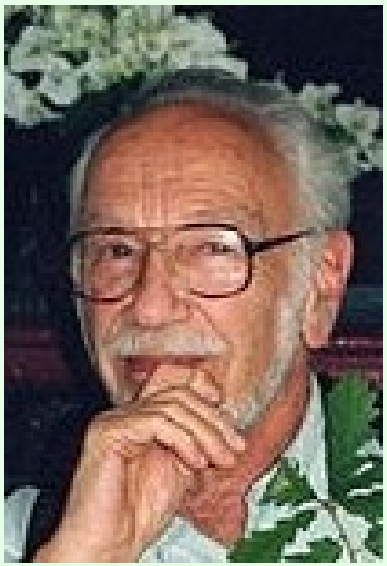
2

...

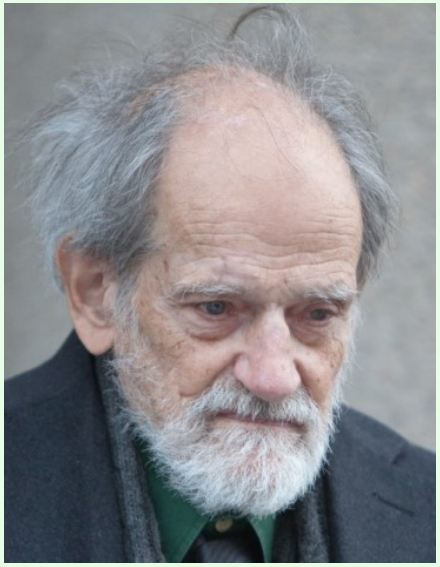
  - Conversely: what if data for ML is generated by strategic agents?
    - Agent being classified herself
    - Data from multiple agents used to find social-welfare maximizing decision

Deferred Acceptance algorithm

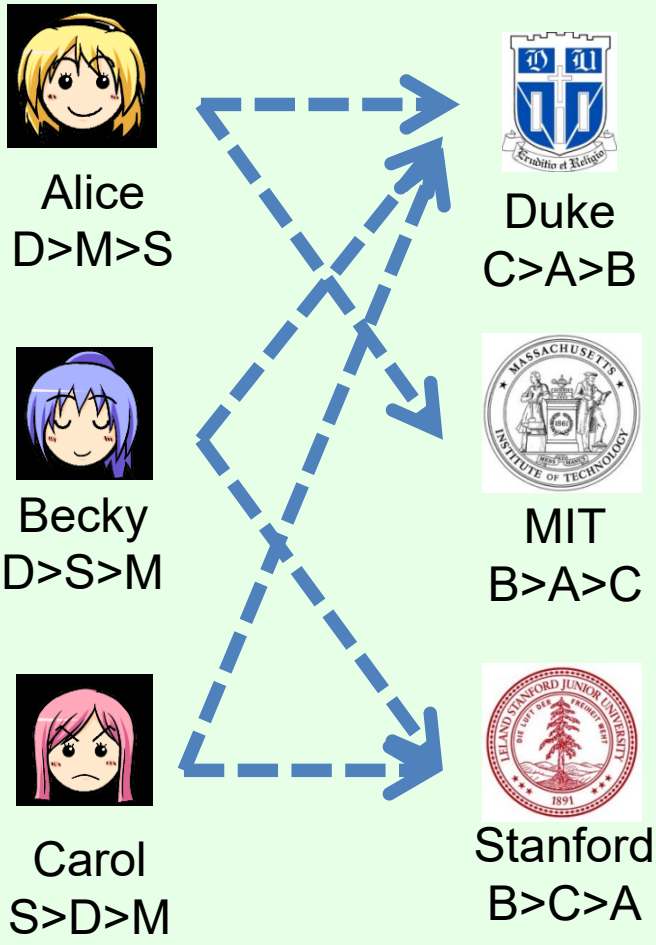
[Gale & Shapley 1962]



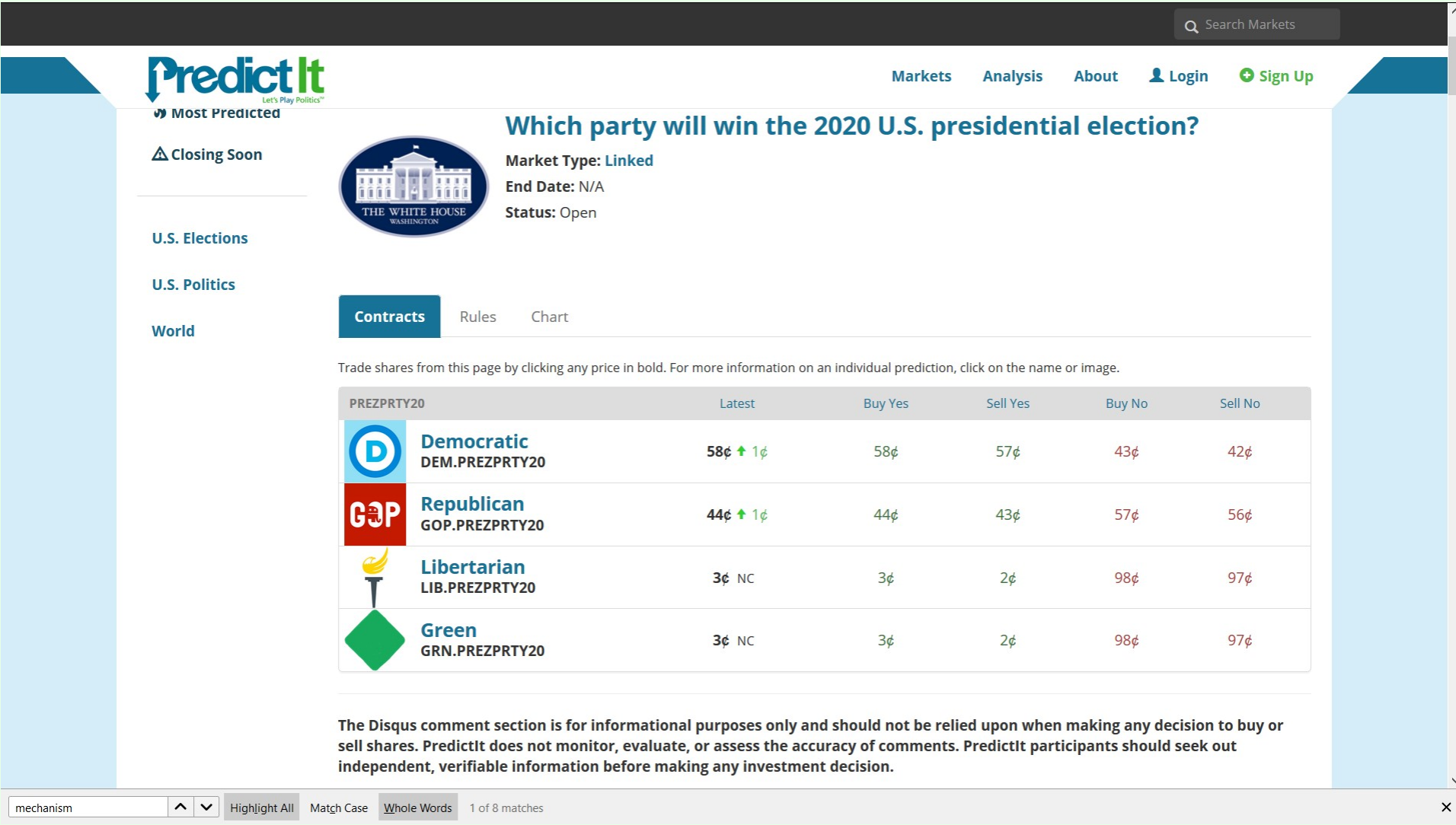
David Gale



Lloyd Shapley



Prediction markets

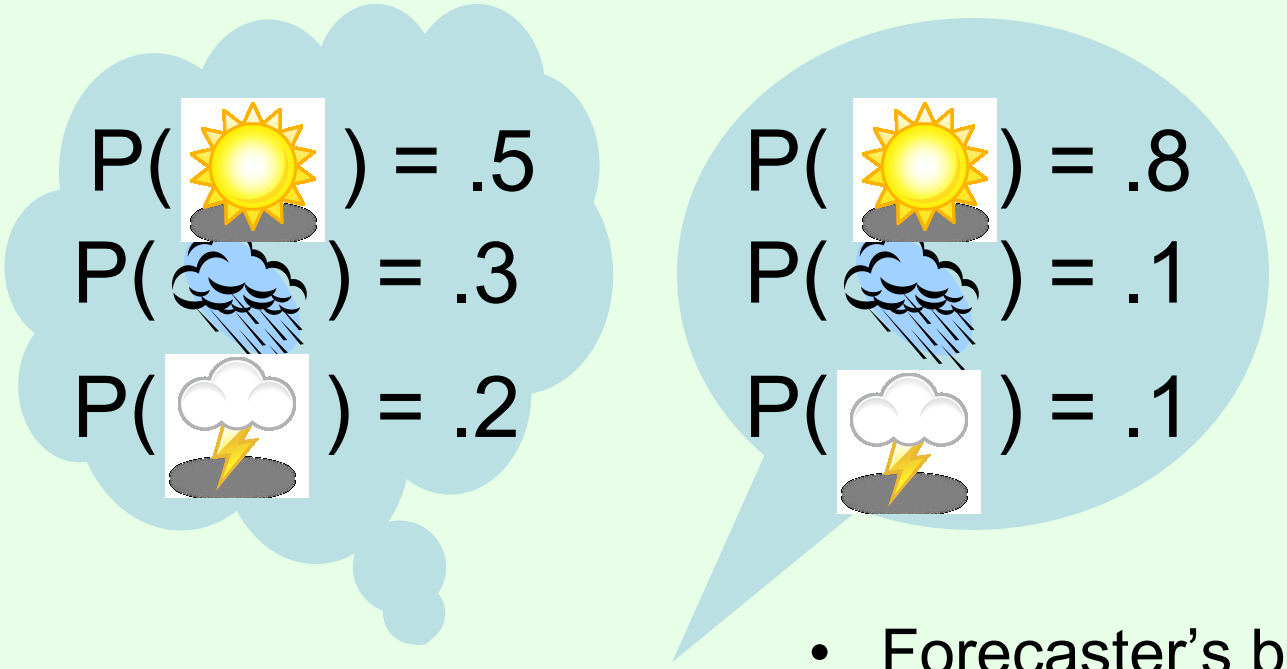


2018/8/27

Financial securities

- Tomorrow there must be one of 
- Agent 1 offers $5 for a security that pays off $10 if  or 
- Agent 2 offers $8 for a security that pays off $10 if  or 
- Agent 3 offers $6 for a security that pays off $10 if 
- Can we accept some of these at offers at no risk?

How to incentivize a weather forecaster



- Forecaster's bonus can depend on
  - Prediction
  - Actual weather on predicted day
- Reporting true beliefs should maximize expected bonus

Peer prediction

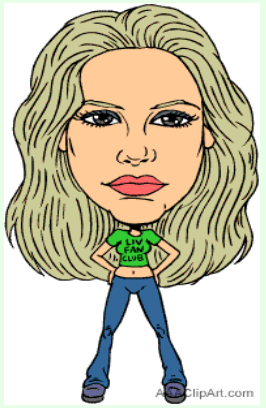
I had a good experience with product X.

5/5

4/5

I had an OK experience with product X.

Each forecaster's bonus depends only on how well it matches the other's



Other kinds of private information in auctions?

my quality estimate

q() = 90

my need

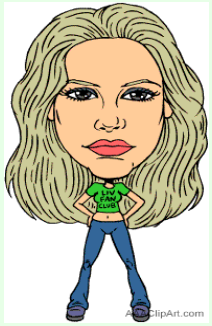
n() = 70

my quality estimate

q() = 80

my need

n() = 75



How should the auctioneer use this information?

Why should economists care about computer science?

- Finding efficient allocations of resources is a (typically hard) computational problem
  - Sometimes beyond current computational techniques
  - If so, unlikely that any market mechanism will produce the efficient allocation (even without incentives issues)
  - Market mechanisms must be designed with computational limitations in mind
  - New algorithms allow new market mechanisms

Why should economists care about computer science…

- Agents also face difficult computational problems in participating in the market
  - Especially acting in a game-theoretically optimal way is often computationally hard
  - Game-theoretic predictions will not come true if they cannot be computed
    - Sometimes bad (e.g., want agents to find right bundle to trade)
    - Sometimes good (e.g., do not want agents to manipulate system)

Why should computer scientists care about economics?

- Economics provides high-value computational problems
- Interesting technical twist: no direct access to true input, must incentivize agents to reveal true input
- Conversely: Computer systems are increasingly used by multiple parties with different preferences (e.g., Internet)
- Economic techniques must be used to
  - predict what will happen in such systems,
  - design the systems so that they will work well
- Game theory is relevant for artificial intelligence
  - E.g., computer poker