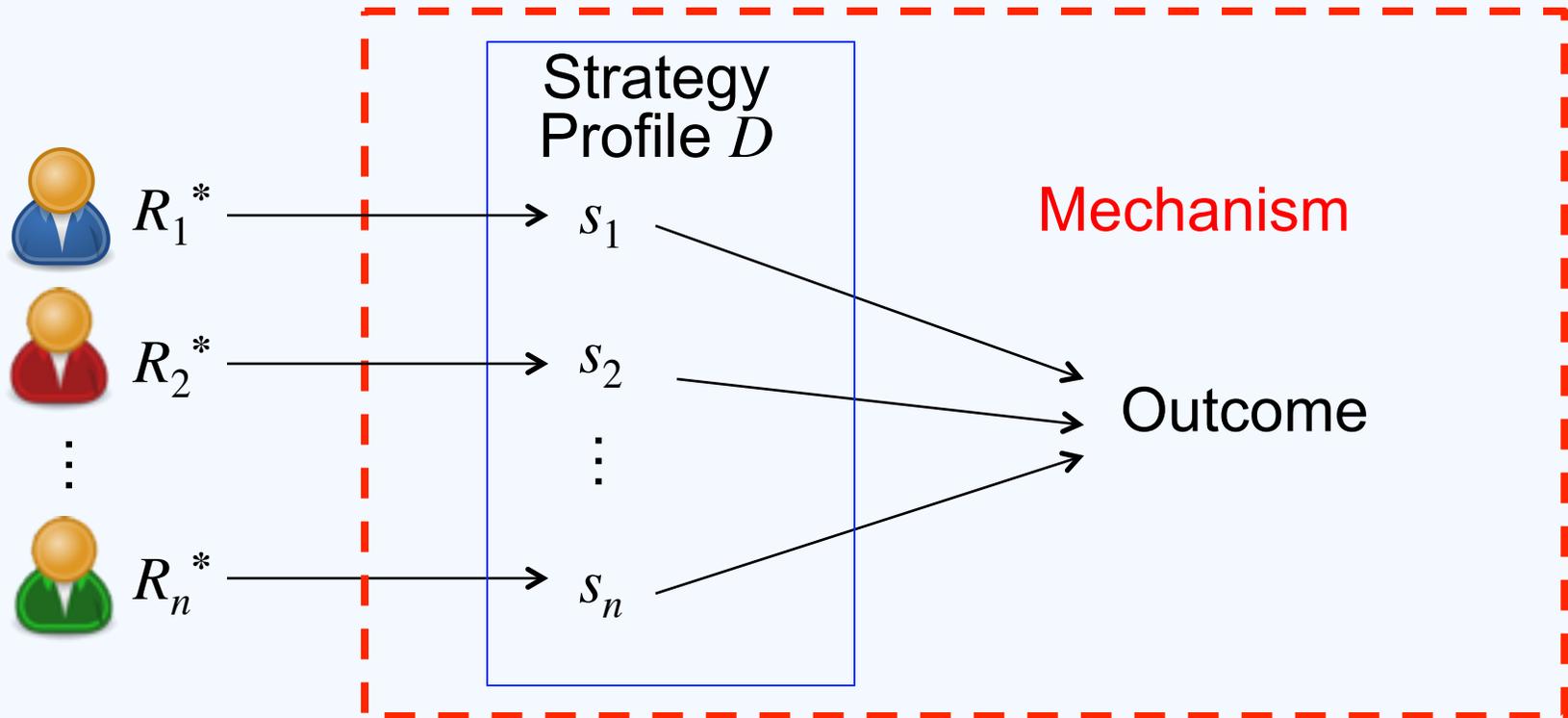


Last class: game theory



- Game theory: predicting the outcome with strategic agents
- Games and solution concepts
 - general framework: NE
 - normal-form games: mixed/pure-strategy NE
 - extensive-form games: subgame-perfect NE

Research 104

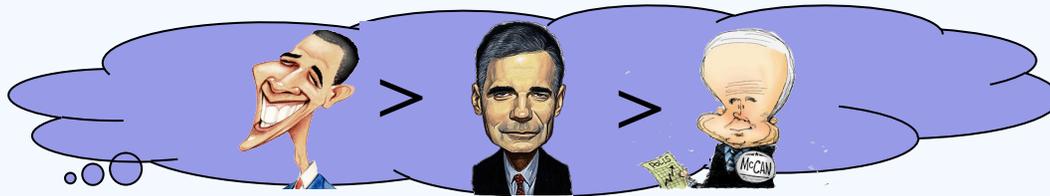
- Lesson 4: scientific skepticism (critical thinking)
 - default for any argument should be “wrong”
 - it is the authors’ responsibility to prove the correctness
 - Once you find an error, try to correct and improve it
- Example: Nash equilibrium in “A beautiful mind”

Election game of strategic voters



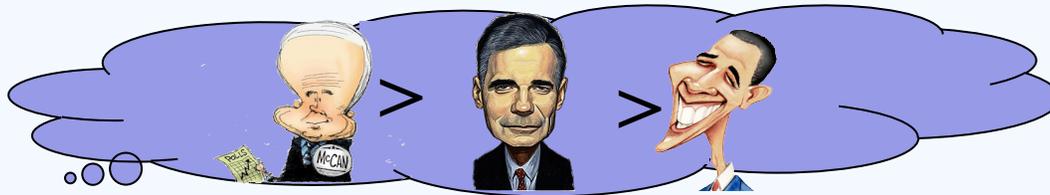
Alice

Strategic vote



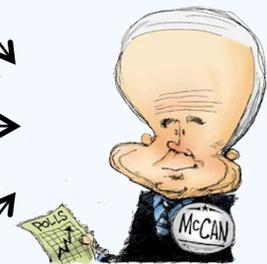
Bob

Strategic vote



Carol

Strategic vote

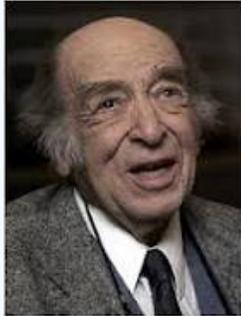


Game theory is predictive

- How to design the “rule of the game”?
 - so that when agents are strategic, we can achieve a designated outcome w.r.t. their **true** preferences?
 - “reverse” game theory
- Example: design a social choice mechanism f so that
 - for **every true** preference profile D^*
 - $\text{OutcomeOfGame}(f, D^*) = \text{Plurality}(D^*)$

Today's schedule: mechanism design

- Mechanism design: Nobel prize in economics 2007



Leonid Hurwicz
1917-2008



Eric Maskin



Roger Myerson

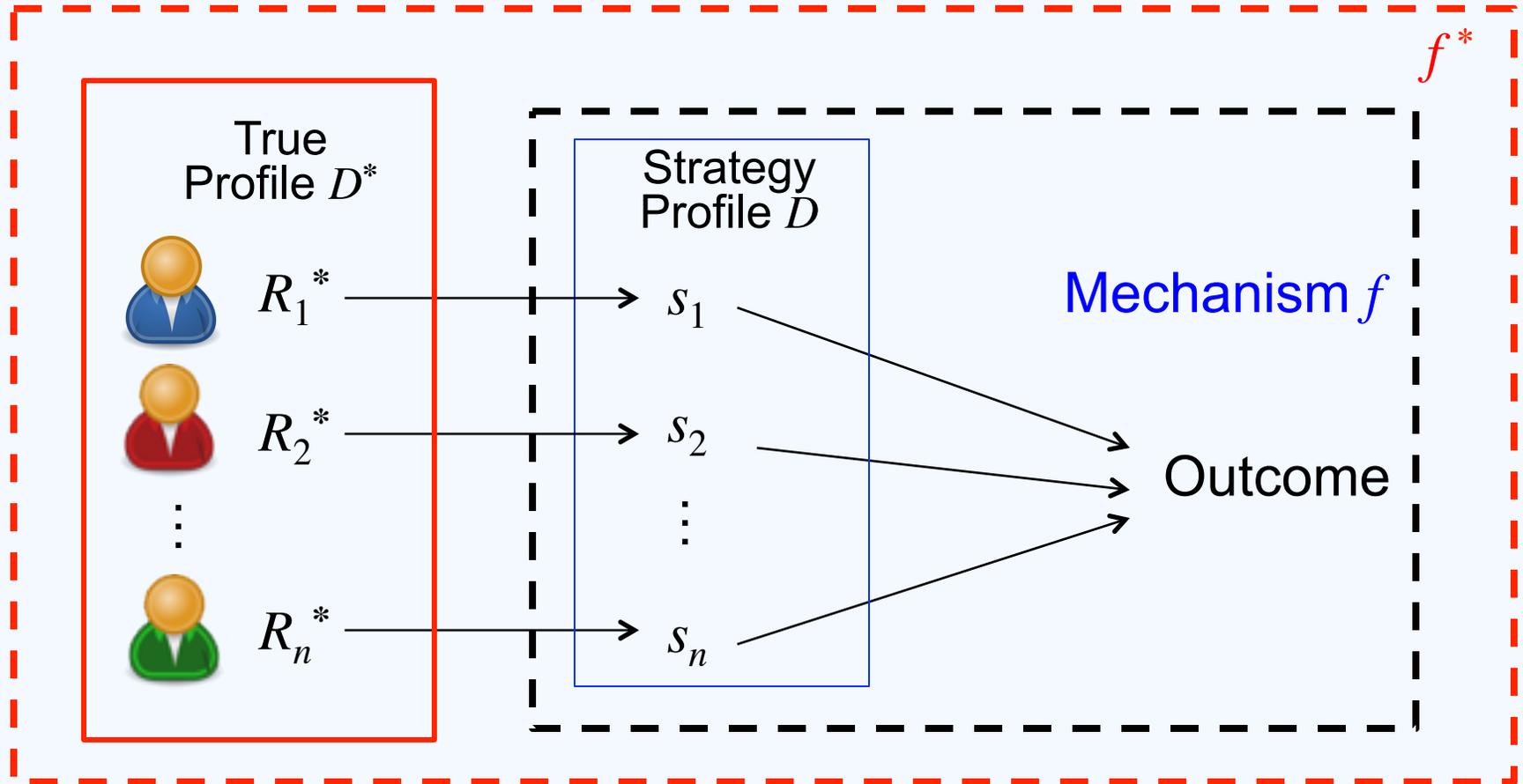
- VCG Mechanism: Vickrey won Nobel prize in economics 1996



William Vickrey
1914-1996

- What? Your homework
- Why? Your homework
- How? Your homework

Implementation



- A game and a solution concept **implement** a function f^* , if
 - for every **true** preference profile D^*
 - $f^*(D^*) = \text{OutcomeOfGame}(f, D^*)$
- f^* is defined for the true preferences

A general workflow of mechanism design

- Pareto optimal outcome
- utilitarian optimal
- egalitarian optimal
- allocation+ payments
- etc

1. Choose a target function f^* to implement

2. Model the situation as a game

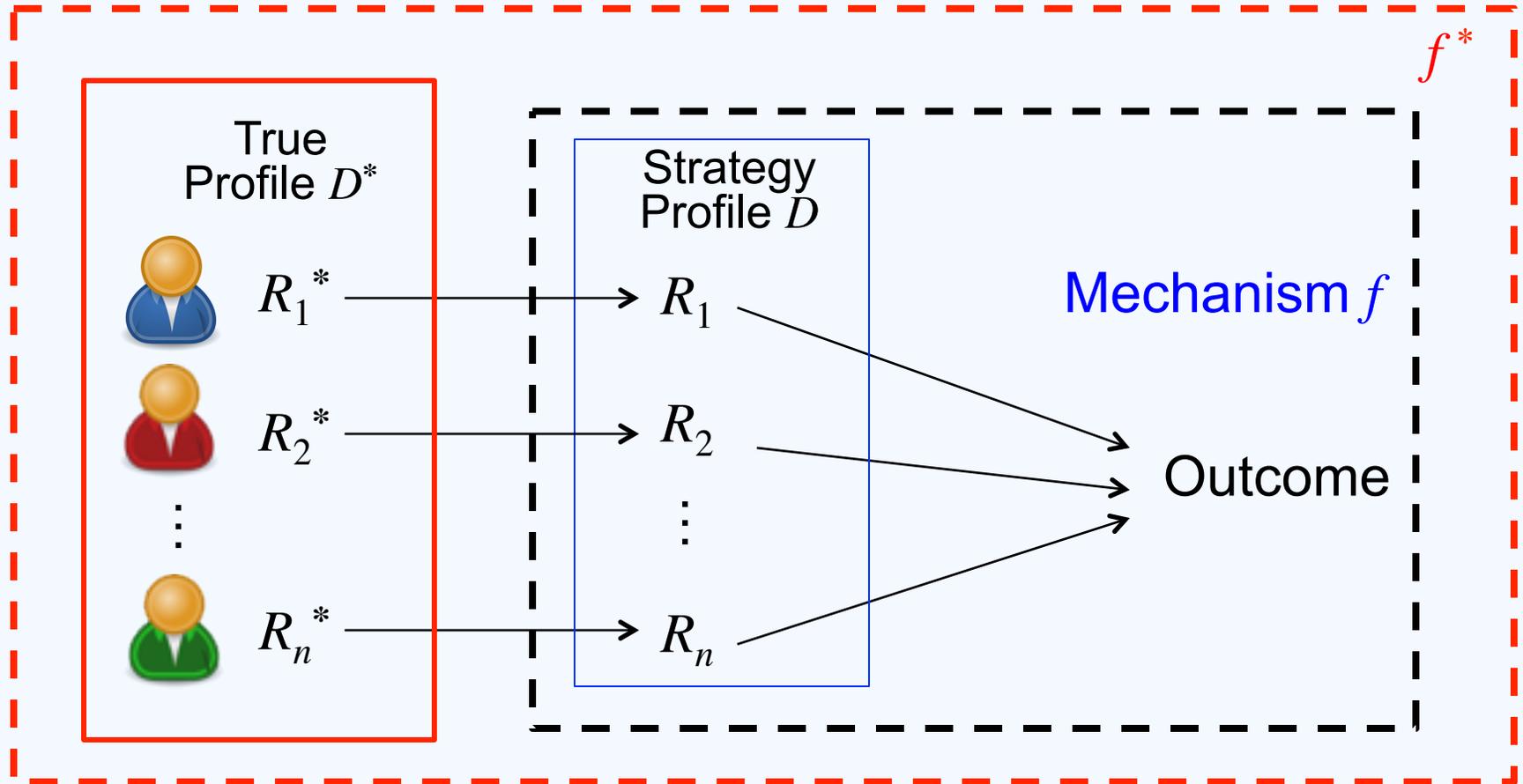
- normal form
- extensive form
- etc

- dominant-strategy NE
- mixed-strategy NE
- SPNE
- etc

3. Choose a solution concept SC

4. Design f such that the game and SC implements f^*

Framework of mechanism design



- Agents (players): $N=\{1,\dots,n\}$
- Outcomes: O
- Preferences (private): total preorders over O
- Message space (c.f. strategy space): S_j for agent j
- Mechanism: $f: \prod_j S_j \rightarrow O$

Frameworks of social choice, game theory, mechanism design

- Agents = players: $N = \{1, \dots, n\}$
- Outcomes: O
- True preference space: P_j for agent j
 - consists of total preorders over O
 - sometimes represented by utility functions
- Message space = reported preference space = strategy space: S_j for agent j
- Mechanism: $f : \prod_j S_j \rightarrow O$

Step 1: choose a target function

(social choice mechanism w.r.t. truth preferences)

- Nontrivial, later after revelation principle

Step 2: specify the game

- Agents: often obvious
- Outcomes: need to design
 - require domain expertise, beyond mechanism design
- Preferences: often obvious given the outcome space
 - usually by utility functions
- Message space: need to design

Step 3: choose a solution concept

- If the solution concept is too weak (general)
 - equilibrium selection
 - e.g. mixed-strategy NE
- If the solution concept is too strong (specific)
 - unlikely to exist an implementation
 - e.g. SPNE
- **We will focus on dominant-strategy NE in the rest of today**

Dominant-strategy NE

- Recall that an NE exists when every player has a **dominant strategy**
 - s_j is a **dominant strategy** for player j , if for every $s_j' \in S_j$,
 1. for every s_{-j} , $f(s_j, s_{-j}) \geq_j f(s_j', s_{-j})$
 2. the preference is strict for some s_{-j}
- A **dominant-strategy NE (DSNE)** is an NE where
 - every player takes a dominant strategy
 - may not exist, but if it exists, then it must be unique

Prisoner's dilemma



Column player



Row player

	Cooperate	Defect
Cooperate	$(-1, -1)$	$(-3, 0)$
Defect	$(0, -3)$	$(-2, -2)$

Defect is the dominant strategy for both players

Step 4: Design a mechanism

Direct-revelation mechanisms (DRMs)

- A special mechanism where for agent j , $S_j = P_j$
 - true preference space = reported preference space
- A DRM f is **truthful (incentive compatible)** w.r.t. a solution concept SC (e.g. NE), if
 - In SC, $R_j = R_j^*$
 - i.e. everyone reports her true preferences
 - **A truthful DRM implements itself!**
- Examples of truthful DRMs
 - always outputs outcome “ a ”
 - dictatorship

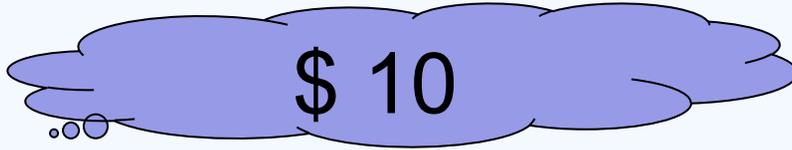
A non-trivial truthful DRM

- Auction for one indivisible item
- n bidders
- Outcomes: { (allocation, payment) }
- Preferences: represented by a **quasi-linear** utility function
 - every bidder j has a private value v_j for the item. Her utility is
 - $v_j - \text{payment}_j$, if she gets the item
 - 0, if she does not get the item
 - suffices to only report a bid (rather than a total preorder)
- Vickrey auction (second price auction)
 - allocate the item to the agent with the highest bid
 - charge her the **second** highest bid

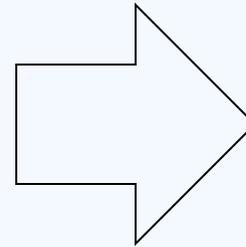
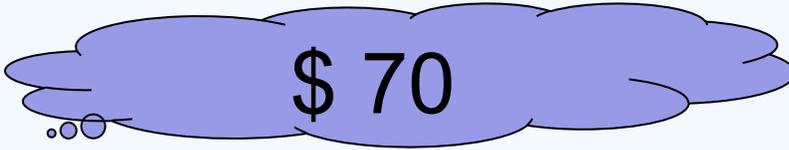
Example



Kyle



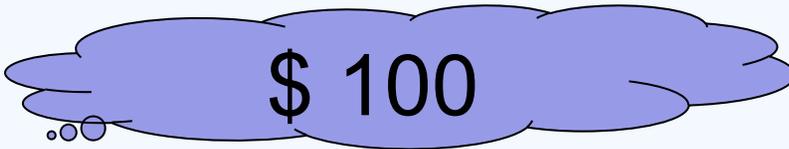
Stan



\$70



Eric



Indirect mechanisms (IM)

- No restriction on S_j
 - includes all DRMs
 - If $S_j \neq P_j$ for **some** agent j , then truthfulness is not defined
 - not clear what a truthful agent will do under IM
- Example
 - Second-price auction where agents are required to report an integer bid

Another example

- English auction

“arguably the most common form of auction in use today” ---wikipedia

- Every bidder can announce a higher price
- The last-standing bidder is the winner
- Implements Vickrey (second price) auction

Truthful DRM vs. IM: usability

- Truthful DRM: f^* is implemented for truthful and strategic agents
 - Truthfulness:
 - if an agent is truthful, she reports her true preferences
 - if an agent is strategic (as indicated by the solution concept), she still reports her true preferences
 - Communication: can be a lot
 - Privacy: no
- Indirect Mechanisms
 - Truthfulness: no
 - Communication: can be little
 - Privacy: may preserve privacy

Truthful DRM vs. IM: easiness of design

- Implementation w.r.t. DSNE
- Truthful DRM:
 - f itself!
 - only needs to check the **incentive conditions**,
i.e. for every j , R_j' ,
 - for every R_{-j} : $f(R_j^*, R_{-j}) \geq_j f(R_j', R_{-j})$
 - the inequality is strict for some R_{-j}
- Indirect Mechanisms
 - Hard to even define the message space

Truthful DRM vs. IM: implementability

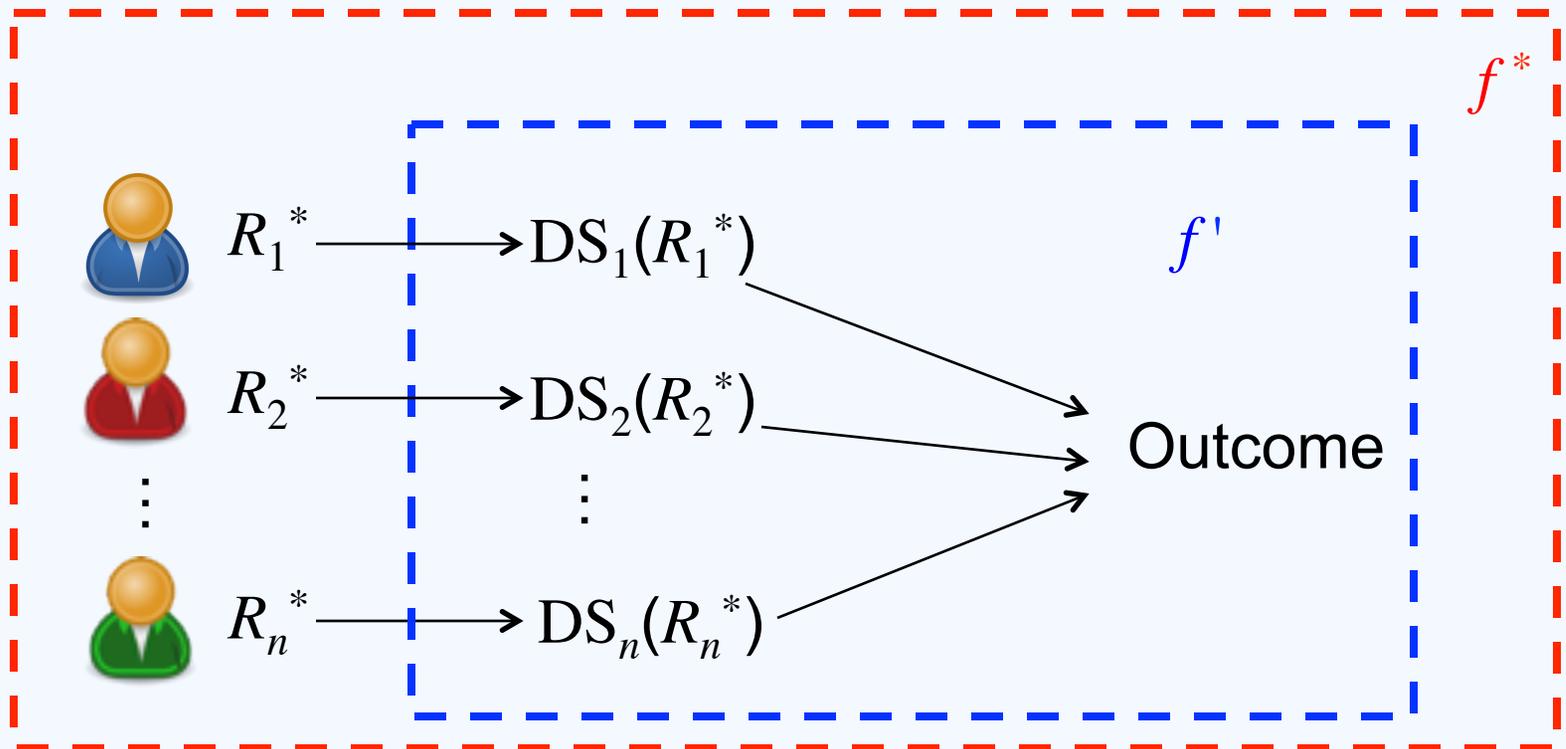
- Can IMs implement more social choice mechanisms than truthful DRMs?
 - depends on the solution concept
- Implementability
 - the set of social choice mechanisms that can be implemented (by the game + mechanism + solution concept)

Revelation principle

- **Revelation principle.** Any social choice mechanism f^* implemented by a mechanism w.r.t. DSNE can be implemented by a truthful DRM (itself) w.r.t. DSNE
 - truthful DRMs is as powerful as IMs in implementability w.r.t. DSNE
 - If the solution concept is DSNE, then designing a truthful DRM implication is equivalent to examining that agents are truthful under f^*
- has a Bayesian-Nash Equilibrium version

Proof

- $DS_j(R_j^*)$: the dominant strategy of agent j
- Prove that f^* is a truthful DRM that implements itself
 - **truthfulness**: suppose on the contrary that f^* is not truthful
 - W.l.o.g. suppose $f^*(R_1, R_{-1}^*) >_1 f^*(R_1^*, R_{-1}^*)$
 - $DS_1(R_1^*)$ is not a dominant strategy
 - compared to $DS_1(R_1)$, given $DS_2(R_2^*), \dots, DS_n(R_n^*)$



Interpreting the revelation principle

- It is a **powerful**, **useful**, and **negative** result
- **Powerful**: applies to any mechanism design problem
- **Useful**: only need to check if truth-reporting is the dominant strategy in f^*
- **Negative**: If any agent has incentive to lie under f^* , then f^* cannot be implemented by any mechanism w.r.t. DSNE

Step 1: Choosing the function
to implement (w.r.t. DSNE)

Mechanism design with money

- Modeling situations with monetary transfers
- Set of **alternatives**: A
 - e.g. allocations of goods
- Outcomes: $\{ (\text{alternative}, \text{payments}) \}$
- Preferences: represented by a **quasi-linear** utility function
 - every agent j has a private value $v_j^*(a)$ for every $a \in A$. Her utility is
$$u_j^*(a, p) = v_j^*(a) - p_j$$
 - It suffices to report a value function v_j

Can we adjust the payments to maximize social welfare?

- Social welfare of a
 - $SCW(a) = \sum_j v_j^*(a)$
- Can any ($\operatorname{argmax}_a SCW(a)$, payments) be implemented w.r.t. DSNE?

The Vickrey-Clarke-Groves mechanism (VCG)

- The Vickrey-Clarke-Groves mechanism (VCG) is defined by

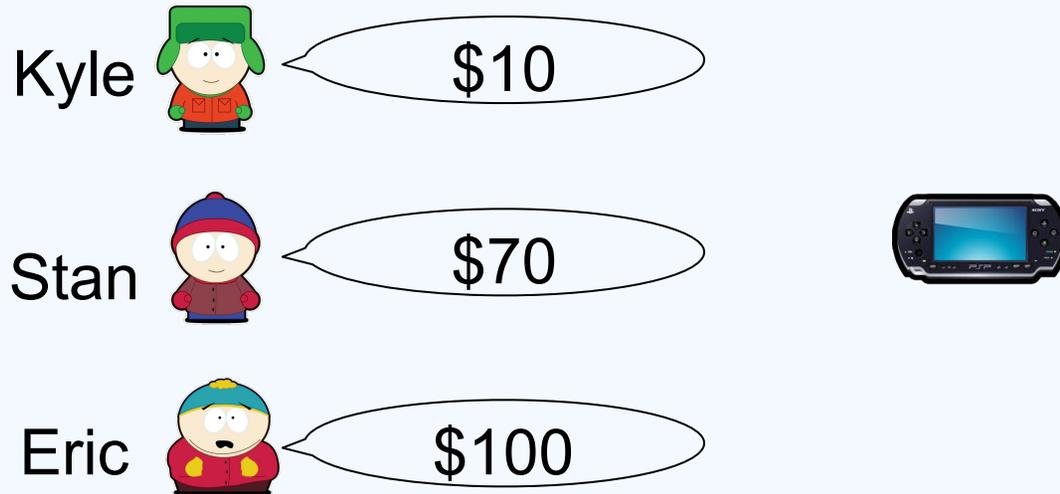
- Alternative in outcome: $a^* = \operatorname{argmax}_a \operatorname{SCW}(a)$

- Payments in outcome: for agent j

$$p_j = \max_a \sum_{i \neq j} v_i(a) - \sum_{i \neq j} v_i(a^*)$$

- **negative externality** of agent j of its presence on other agents
- Truthful, efficient
- A special case of **Groves mechanism**

Example: auction of one item



- Alternatives = (give to K, give to S, give to E)
- $a^* =$ 
- $p_1 = 10 - 10 = 0$
- $p_2 = 10 - 10 = 0$
- $p_3 = 7 - 0 = 7$

Wrap up

- Mechanism design:
 - the social choice mechanism f^*
 - the game and the mechanism to implement f^*
- The revelation principle: implementation w.r.t. DSNE = checking incentive conditions
- VCG mechanism: a generic truthful and efficient mechanism for mechanism design with money

Looking forward

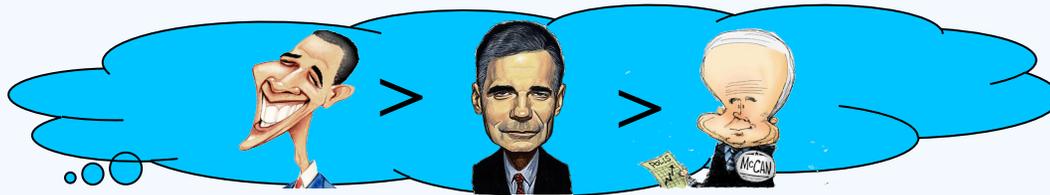
- The end of “pure economics”
 - Social choice: 1972 (Arrow), 1998 (Sen)
 - Game theory: 1994 (Nash, Selten and Harsanyi), 2005 (Schelling and Aumann)
 - Mechanism design: 2007 (Hurwicz, Maskin and Myerson)
 - Auctions: 1996 (Vickrey)
- The next class: introduction to computation
 - Linear programming
 - Basic computational complexity theory
- Then
 - Computation + Social choice
- **HW1 is due on Thursday before class**

NE of the plurality election game

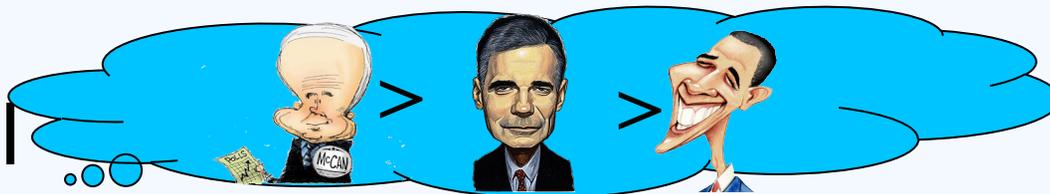
YOU



Bob



Carol



Plurality rule



- Players: $\{ \text{YOU}, \text{Bob}, \text{Carol} \}$, $n=3$
- Outcomes: $O = \{ \text{Obama}, \text{Clinton}, \text{McCain} \}$
- Strategies: $S_j = \text{Rankings}(O)$
- Preferences: $\text{Rankings}(O)$
- Mechanism: the plurality rule

Proof (1)

- Given
 - f^* implemented by f' w.r.t. DSNE
- Construct a DRM f that “**simulates**” the strategic behavior of the agents under f' , $DS_j(u_j)$

$$f(u_1, \dots, u_n) = f'(DS_1(u_1), \dots, DS_n(u_n))$$

