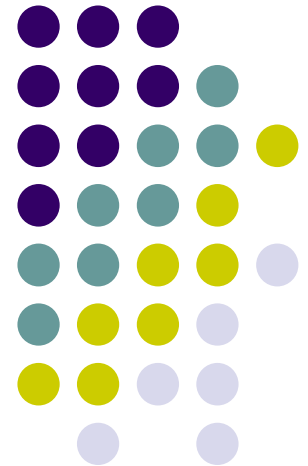
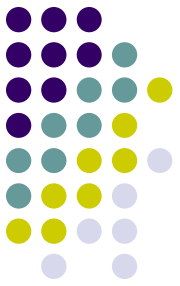


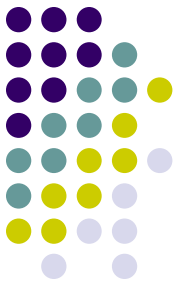
Assignment





House allocation problems

- In some matching markets, only one side of the market has preferences (or we care mostly about the preferences of one side).
- Examples
 - students picking housing on campus
 - students picking freshman seminars
 - sports teams picking players out of college



House allocation problem

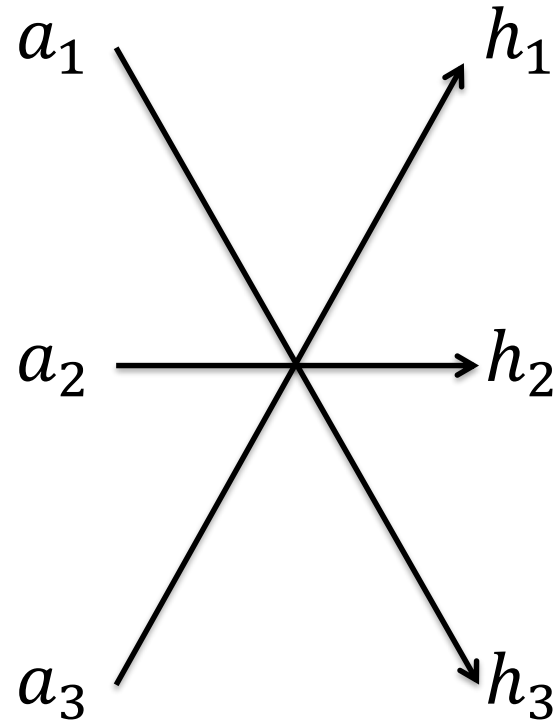
- N agents and N houses
- Each agent has strict preferences over houses.
- Goal: assign each agent to a house.
- What would be a “good” outcome?



Desirable Allocation?

3 agents a_1, a_2, a_3 and
3 houses h_1, h_2, h_3

- $P(a_1): h_1 > h_2 > h_3$
- $P(a_2): h_3 > h_2 > h_1$
- $P(a_3): h_1 > h_3 > h_2$





Efficient Allocation

- A (*Pareto*) *efficient* allocation is one that exhausts all gains from trade –
=> i.e. there is no alternative allocation that makes all better off and at least one strictly better off.
⇒ Any improvement attempt must make someone worse off

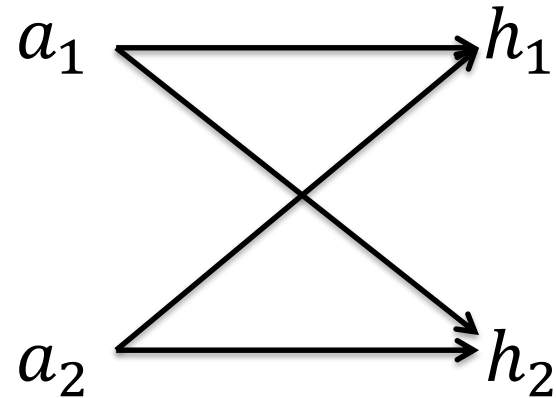
Individual Rationality: No worse assignment than own house.



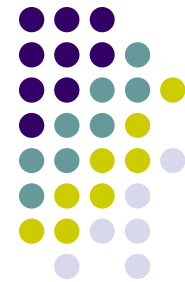
Efficient Allocations

3 agents a_1, a_2, a_3 and
3 houses h_1, h_2, h_3

- $P(a_1): h_1 > h_2 > h_3$
- $P(a_2): h_3 > h_2 > h_1$
- $P(a_3): h_1 > h_3 > h_2$



(Random) Serial Dictatorship



- Each agent gets a priority (perhaps randomly assigned as in the housing draw).
- Agents pick houses in order of their priority.

Theorem. The serial dictatorship is (ex post) efficient (i.e. no mutually agreeable trades afterwards) and strategy-proof.



Proof

Strategy-proofness:

- Agent with first pick gets her preferred house, so clearly no incentive to lie.
- Agent with second pick gets her preferred house among remaining houses, so again no reason to lie.
- and so on...

Proof



Efficiency:

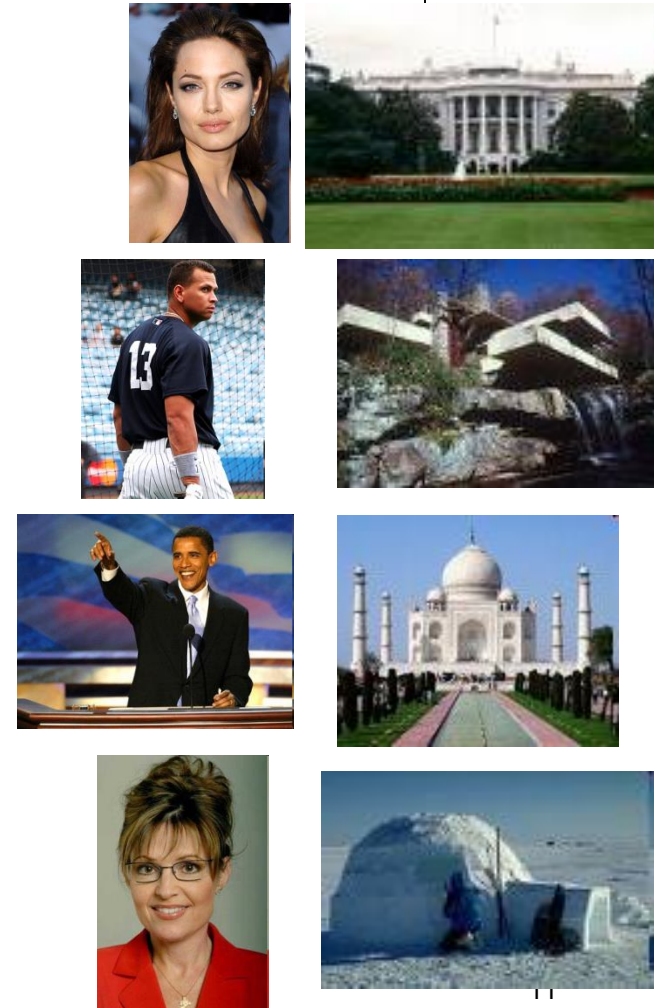
- Agent with priority one doesn't want to trade.
- Given that she is out, agent with priority two doesn't want to trade.
- And so on....



Top Trading Cycles

- Now imagine that agents start with a house, but the original allocation might not be efficient.
- **Gale's TTC algorithm**
 - Each person points to most preferred house
 - Each house points to its owner
 - This creates a directed graph, with at least one cycle.
 - Remove all cycles, assigning people to the house they are pointing at.
 - Repeat using preference lists where the assigned houses have been deleted.

A decorative graphic in the bottom right corner consisting of a grid of colored dots in shades of purple, teal, yellow, and light blue, arranged in a pattern that tapers to the right.



House allocation from endowments :The Core



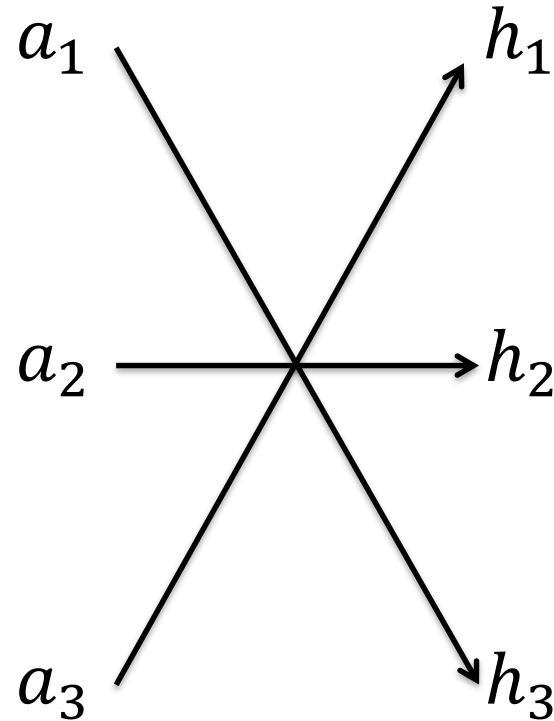
- Consider a candidate assignment in the house allocation problem:
 - A coalition of agents **blocks** if, from their initial endowments, there is an assignment among themselves that they all prefer to the candidate assignment.
- The **core** consists of all feasible unblocked assignments.



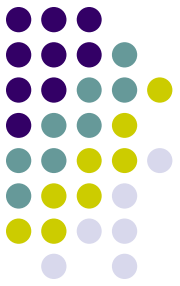
Core Allocation?

3 agents a_1, a_2, a_3 and
3 houses h_1, h_2, h_3

- $P(a_1): h_1 > h_2 > h_3$
 - $P(a_2): h_3 > h_2 > h_1$
 - $P(a_3): h_1 > h_3 > h_2$
-
- a_i owns h_i for $i = 1, 2, 3$



Properties of TTC



Theorem. The outcome of the TTC algorithm is the unique core assignment in the housing market.



Proof

Core

- Blocking coalition cannot involve only those matched at round one (all agents get first choice).
- Blocking coalition cannot involve only those matched in first two rounds (can't improve round one guys, and to improve round two guys, need to displace round one guy).
- And so on by induction.



Proof

Uniqueness:

- Consider doing something other than assigning the round one individuals their TTC houses. They would get together and block.
- Fixing the assignments for the individuals cleared at round one of the TTC, consider an assignment that differs for the individuals that would be assigned at round 2 of TTC.
- Same argument applies. And inductively for rounds 3,4....



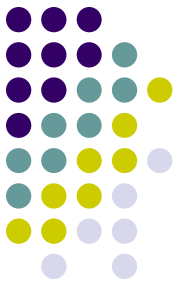
Incentives in the TTC

Theorem. The TTC algorithm is strategy-proof.

Proof. For any agent assigned at round n if truthful

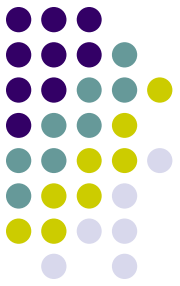
- No change in his report can give him a house that was assigned in earlier rounds.
- No house assigned in a later round will make him better off.
- So no benefit to doing anything but reporting truthfully.

Combining the problems: On-campus housing



- What if some individuals start with houses but some do not?
- A common problem in allocating student housing
 - Many universities, e.g. Caltech, CMU, Duke, Michigan, Northwestern, Penn use a variation of *random serial dictatorship*.
- Let's see how it works.

House allocation with existing tenants



- Problem components
 - newcomers
 - existing tenants
 - priority order
- Main application: Graduate housing

Examples: Michigan, Princeton, Rochester, Stanford, CMU, MIT, etc.

What is a good mechanism?



1. **Individual rationality** (existing tenants)
2. **Fairness** (priority order)
3. **Efficiency** (e.g. Pareto)
4. **Incentive compatibility** (no gaming)

The Model



- Agents: $I = \{1, 2, \dots, n\}$
 - Existing tenants: I_E
 - Newcomers: I_N
- Houses $H = \{h_1, h_2, \dots, h_m\}$
 - Occupied houses: I_O
 - Vacant houses: I_V
- A list of strict preferences $R = (R_i)_{i \in I}$
- A priority order $f: \{1, \dots, n\} \rightarrow I$



- A **house allocation problem with existing tenants** is a pair consisting of
 - List of agents' preferences (R)
 - A priority order (f)
- An **allocation** is a list s.t.
 - every agent is assigned at most one house
 - no house is assigned to more than one agent

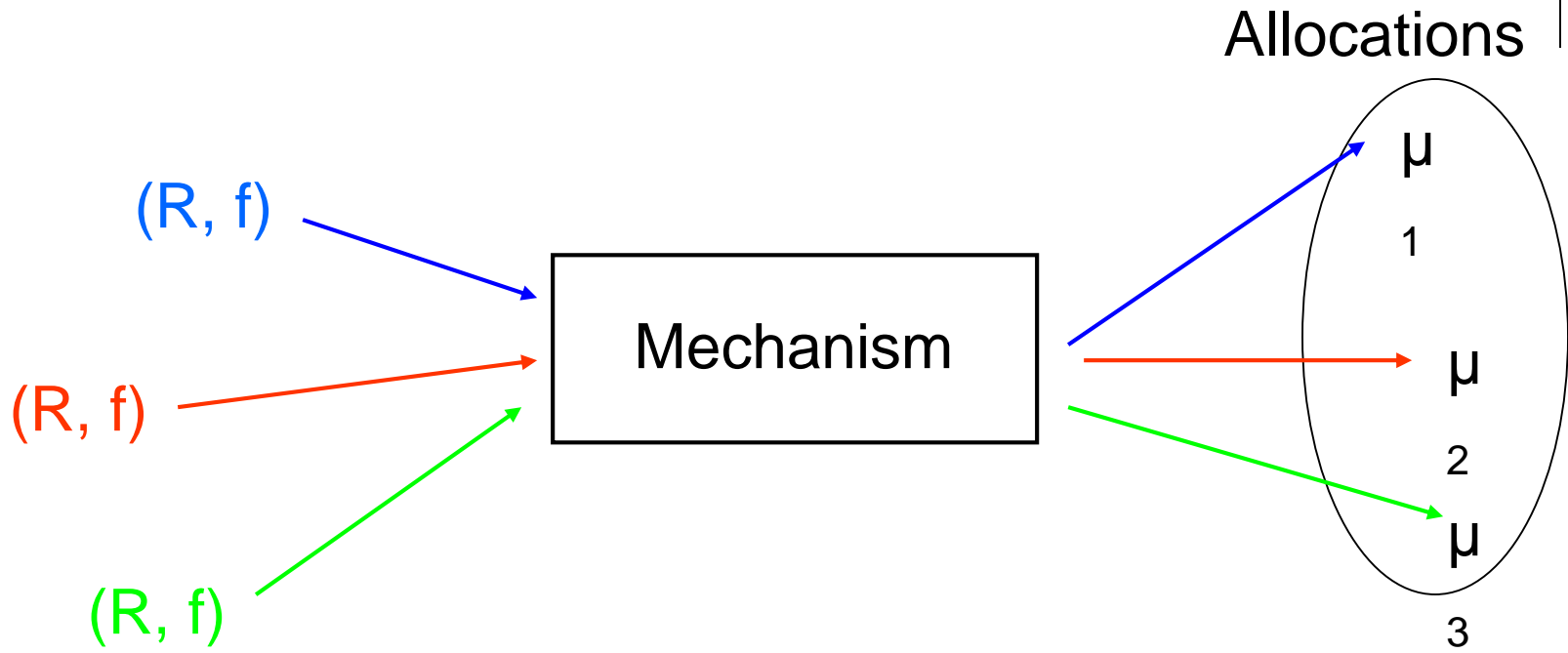


Example:

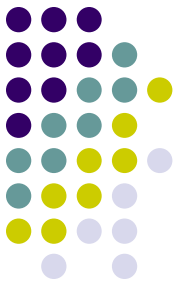
$$I_E = \{1, 2\}, \quad I_N = \{3, 4\}, \quad H_O = \{h_1, h_2\}, \quad H_V = \{a, b, c\}$$

R_1	R_2	R_3	R_4	f
b	c	a	h_2	4
h_1	a	b	c	2
a	h_2	h_1	h_1	5
h_2	h_1	c	a	3
c	b	h_2	b	1

What is a mechanism?



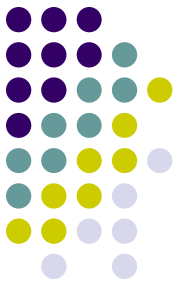
Properties of Mechanisms



1. **Individual Rationality:** No existing tenant is assigned a house which is worse for him than his current house.

i.e., for all R , all $i \in I_E$, $\varphi_i(R) R_i h_i$

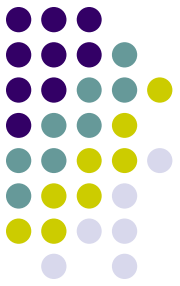
Properties of Mechanisms



2. **Fairness:** An agent prefers someone else's assignment (to his own) only if either of the following holds:

- The other agent is an existing tenant who is assigned his own house
- The other agent has higher priority

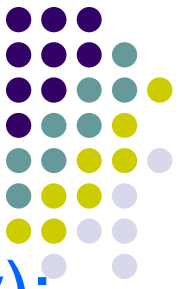
Properties of Mechanisms



3. **Pareto Efficiency:** It is not possible to find an alternative allocation that makes

- All agents at least as well off
- At least one agent strictly better off

However, **an inefficient mechanism need not always select inefficient outcomes!!!**



Properties of Mechanisms

4. Strategy-proofness (Incentive compatibility):

It is always a dominant strategy for each agent to truthfully reveal his preferences.

i.e., for all R , all $i \in I$, and all R'_i : $\varphi_i(R) R_i \varphi_i(R'_i, R_{-i})$

Example: Boston's old school system

Real-life Mechanisms

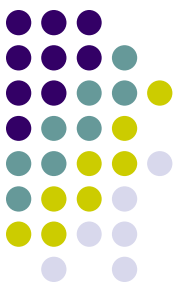


1. Random serial dictatorship with squatting rights

(CMU, Duke, Harvard, Northwestern, Upenn, etc.)

- Each existing tenant initially decides whether to participate or not. If participates, gives up his current house
- A priority ordering f of participants is randomly chosen
- First agent (according to f) is assigned his favorite house, second agent is assigned his favorite house among the remaining houses, and so on.

Random serial dictatorship with squatting rights



Properties

1. Individual rationality ✗
2. Fairness ✓
3. Pareto efficiency ✗
4. Incentive compatibility ✓

Real-life Mechanisms



2. MIT-NH4 Mechanism

1. The first agent is *tentatively* assigned his top choice among all houses, the next agent is *tentatively* assigned his top choice among the remaining houses, and so on, until a **squatting conflict** occurs.
2. A **squatting conflict** occurs if it is the turn of an existing tenant but every remaining house is worse than his current house. That means someone else, the **conflicting agent**, is *tentatively* assigned the existing tenant's current house. When this happens, solve the **squatting conflict** as follows:
 - Assign the existing tenant his current house and remove him
 - Erase all tentative assignments starting after the **conflicting agent**
3. The process is over when there are no houses or agents left.



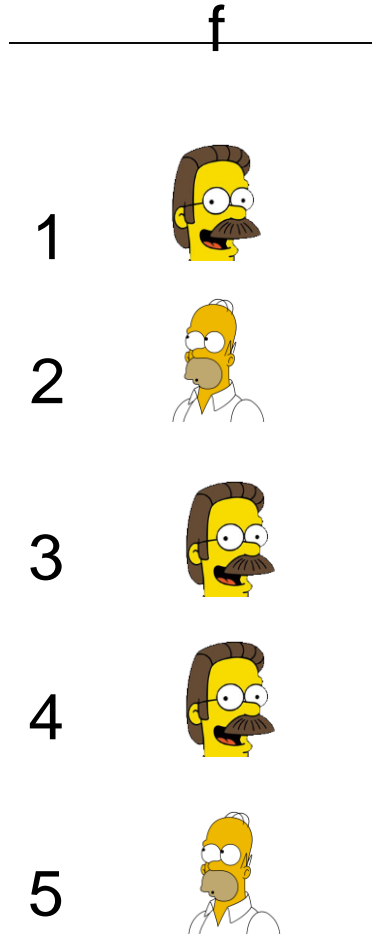
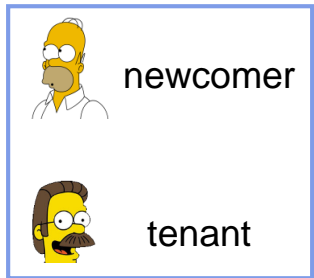
Example:

$$I_E = \{1, 3, 4\}, \quad I_N = \{2, 5\},$$

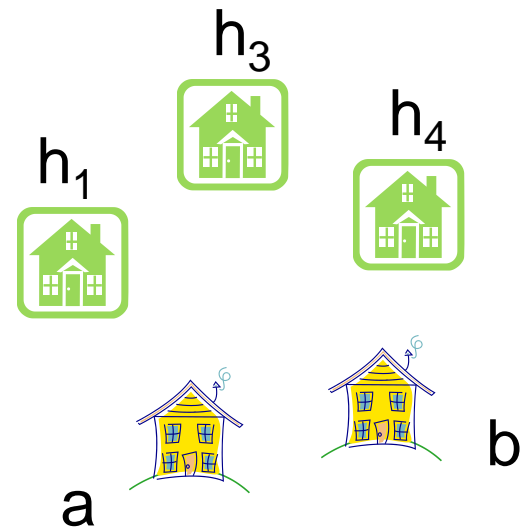
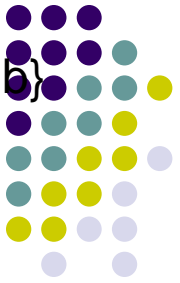
$$H_O = \{h_1, h_3, h_4\}, \quad H_V = \{a, b\}$$

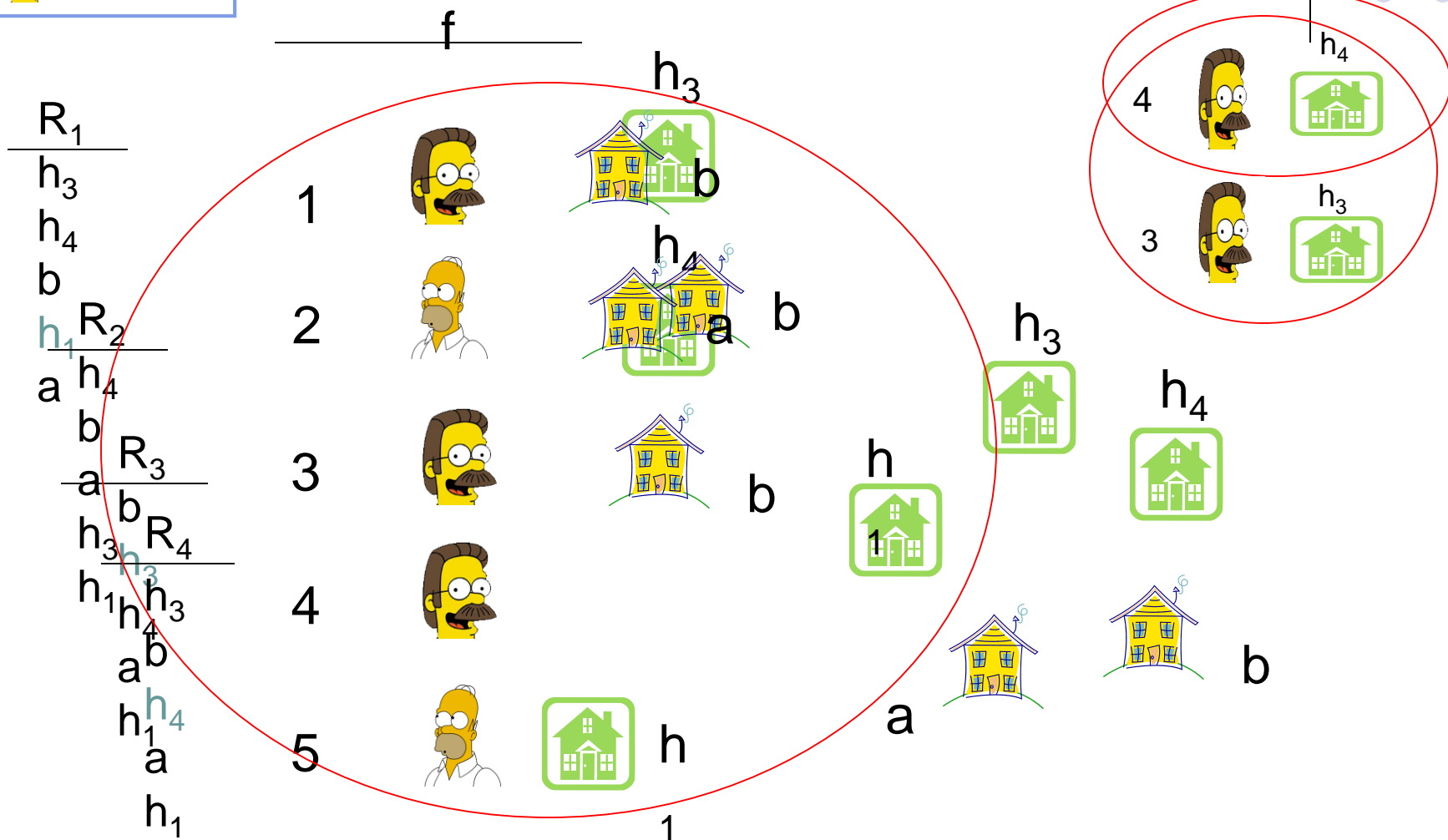
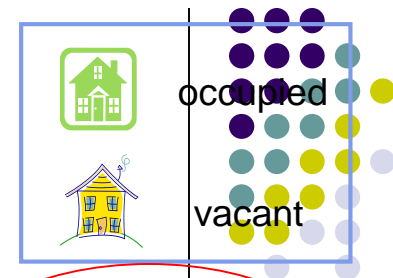
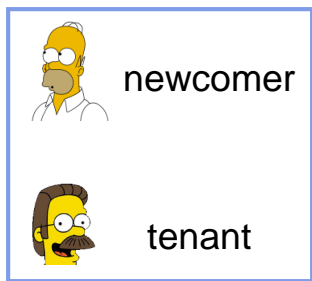
R_1	R_2	R_3	R_4	R_5	f
h_3	h_4	b	h_3	h_4	1
h_4	b	h_3	b	b	2
b	a	h_4	h_4	h_3	3
h_1	h_3	a	a		4
h_1	a	h_1	h_1	h_1	5
a					

$$I_E = \{1, 3, 4\}, \quad I_N = \{2, 5\},$$



$$H_O = \{h_1, h_3, h_4\}, \quad H_V = \{a, b\}$$





MIT-NH4 Example: Final outcome



R_1	R_2	R_3	R_4	R_5
h_3	h_4	b	h_3	h_4
h_4	b	h_3	b	b
b	a	h_4	h_4	h_3
h_1	h_3	a	a	
h_1	a	h_1	h_1	h_1
a				



MIT-NH4 Mechanism

Proposition 2:

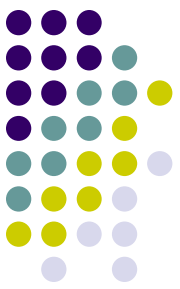
1. Individual rationality ✓

2. Fairness ✓

3. Pareto efficiency ✗

4. Incentive compatibility ✓

A mechanism from recent theory

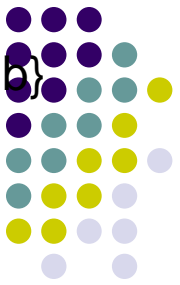



3. TTC Mechanism


- Assign the first agent (according to f) his top choice, the second agent his top choice among the remaining houses, and so on, until someone demands the house of an existing tenant.
 - If at that point the existing tenant whose house is demanded is already assigned a house, then do not disturb the procedure.
 - Otherwise insert him to the top and proceed. Similarly, insert any existing tenant who is not already served at the top of the line once his or her house is demanded.
 - If at any point, a loop forms, (it is formed by exclusively existing tenants and each of them demands the house of the tenant next in the loop), remove all agents in the loop by assigning them the houses they demand, and proceed.

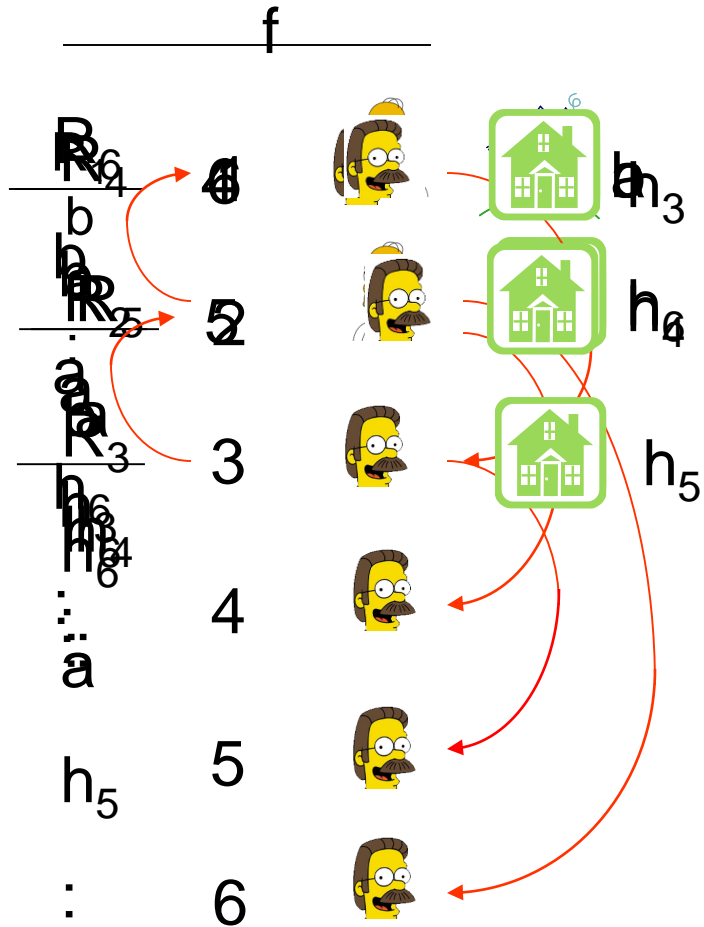
$$I_E = \{3, 4, 5, 6\}, \quad I_N = \{1, 2\},$$

$$H_O = \{h_3, h_4, h_5, h_6\}, \quad H_V = \{a, b\}$$



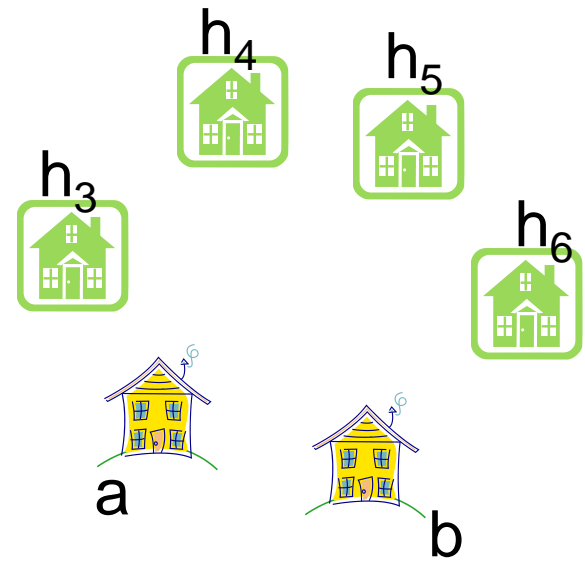

newcomer


tenant




occupied


vacant



Top Trading Cycles Mechanism



Properties

1. Individual rationality



2. Fairness



3. Pareto efficiency

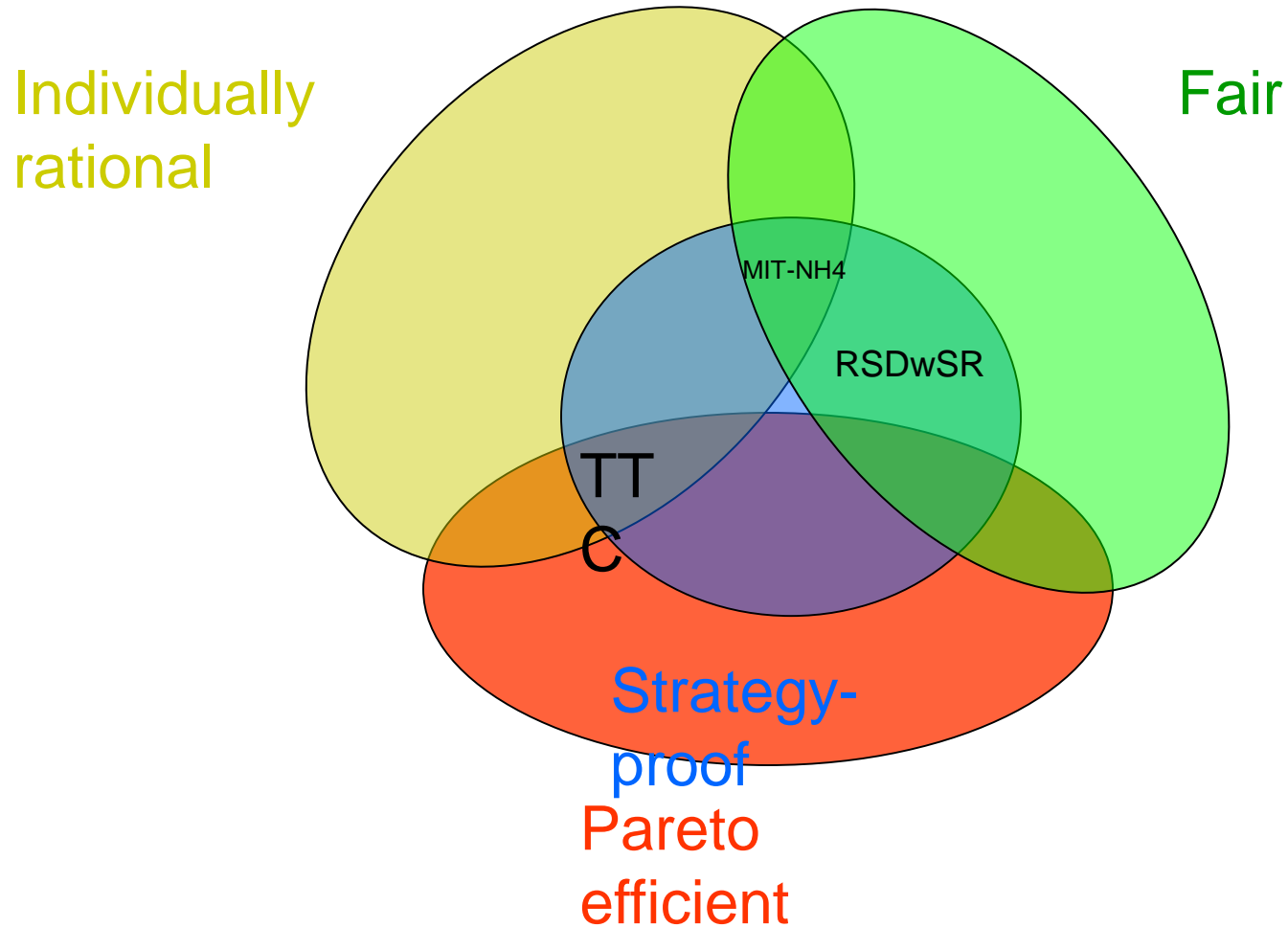


4. Incentive compatibility





SUMMARY



Motivating Question



Given the nice features of GSSM, can we also find a way to also use it for **house allocation**?

Answer: Two issues

1. Single priority order



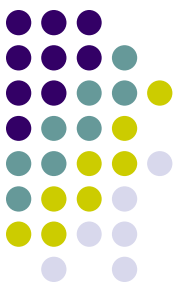
Use the same priority order

2. Individual rationality



Insert the existing tenant to the top of the order

Going back to house allocation

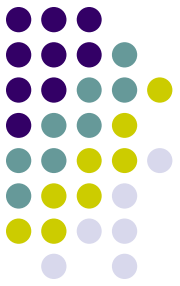


Proposition 3:

4. Modified Gale-Shapley Mechanism

1. Individual rationality ✓
2. Fairness ✓
3. Pareto efficiency ✗
4. Incentive compatibility ✓

An interesting coincidence

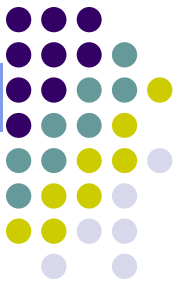


Theorem 1: The MIT-NH4 mechanism and the modified Gale-Shapley mechanism are equivalent (i.e., they always give the same outcome).

Going back in history: Roth (1984) had showed that

NRMP matching mechanism since 1951 = Gale-Shapley

The best **fair** and **individually rational** mechanism

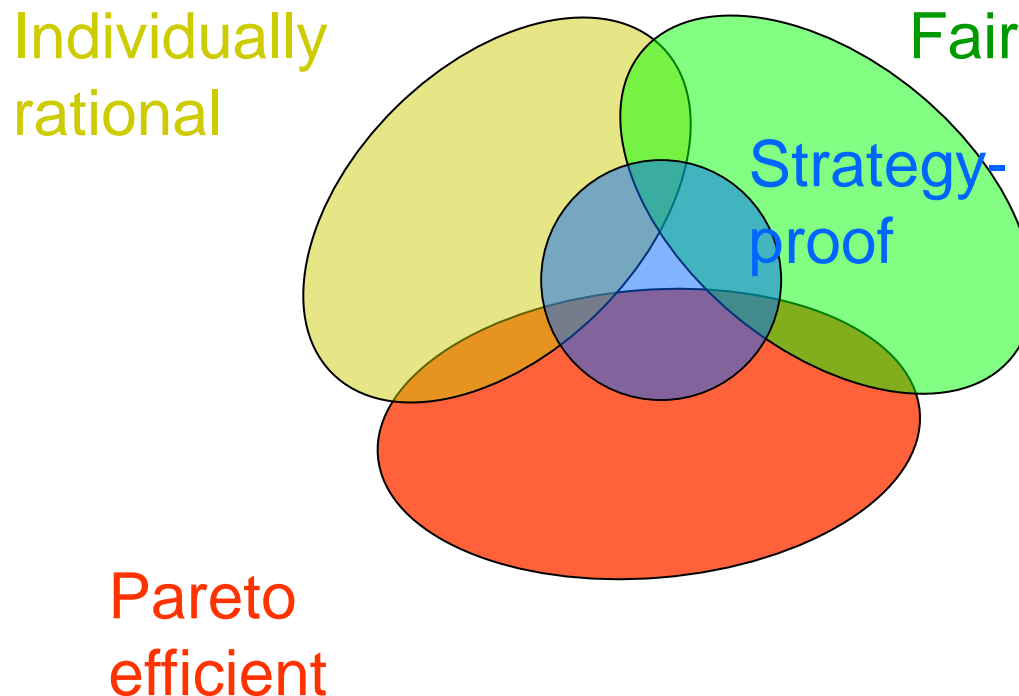


Corollary: The MIT-NH4 mechanism (as well as the modified Gale-Shapley mechanism) Pareto dominates any other **fair** and **individually rational** mechanism.

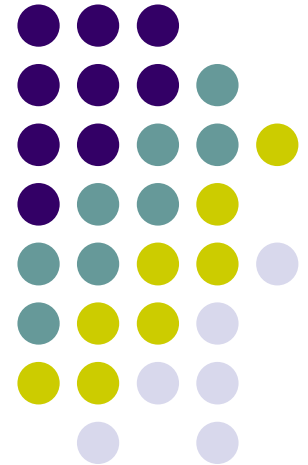
Trade-offs between properties



Proposition 1: There is no mechanism which is individually rational, fair, and Pareto efficient.



Kidney Exchange





Kidney Exchange

- Transplants are standard treatment for patients with failed kidneys.
 - Shortage of kidneys
 - Over 101,170 patients in the waitlist.
 - 3000 patients added each month
- Some statistics from 2013
 - 16,896 transplants from deceased donors
 - 5733 transplants from living donors
 - 4453 patients died on the waiting list.



Resolving the shortage

- Buying and selling kidneys - illegal.
- Section 301 of National Organ Transplant Act
 - “it shall be unlawful for any person to knowingly acquire, receive or otherwise transfer any human organ for valuable consideration for use in human transplantation.”
- Increase cadaveric kidneys (e.g. make donation the default).
- Focus: increasing live donor kidneys.



Donor Kidneys

- Deceased donors: a centralized mechanism has long been in use – prioritizing patients higher on the waitlist.
- Living donors: mostly friends and relatives of a patient – numbers have been increasing

	1988	1998	2008
All donors	5,693	9,761	10,920
Deceased	3,876	5,339	5,992
Live	1,817	4,422	4,928



Compatibility

- Donor kidney must be compatible with patient
- Blood type match
 - O type patients can receive O kidneys
 - A type patients can receive O or A kidneys
 - B type patients can receive O or B kidneys
 - AB type patients can receive any blood type kidney
- Also tissue type match (HLA compatibility).
- Potential inefficiency: if a patient has a donor but can't use the donor's kidney, the donor goes home.



Paired Exchange

- Paired exchange: match two donor-patient pairs...
 - Donor 1 is compatible with Patient 2, not Patient 1
 - Donor 2 is compatible with Patient 1, not Patient 2
- List exchange: match one incompatible donor-patient pair and the waiting list
 - Donor of incompatible pair donates to patient at the top of the waiting list.
 - Patient of incompatible pair goes to the top of the wait list.
- Altruistic (good Samaritan) donations.



Do we know this problem?

- Problem seems very similar to house allocation with existing tenants.
- Roth, Sonmez and Unver (2004, *QJE*)
 - The problems are (essentially) equivalent
 - TTC can be used to efficiently assign kidneys.
- In 2004, RSU and doctors in Boston established first clearinghouse for New England.

Release Date:
12/09/2010

First Kidney Paired Donor Transplants Performed as Part of National Pilot Program

Ken Crowder of St. Louis and Kathy Niedzwiecki of Pelham, N.H., are experiencing renewed life and health thanks to the generosity of two living kidney donors

Rebecca Burkes of St. Louis had intended to be a living donor for her fiancé, Mr. Crowder, and Cathy Richard of Henniker, N.H., had planned to donate to her sister-in-law, Ms. Niedzwiecki -- only to find that both were medically incompatible with their intended recipient. But in the first paired donation arranged through a national pilot program of the Organ Procurement and Transplantation Network (OPTN), Ms. Burkes was able to donate to Ms. Niedzwiecki and Ms. Richard became a donor for Mr. Crowder.

"Paired donation is helping the transplant community help people who otherwise could not get a living donor transplant. We're proud to be able to coordinate these for the first time using a national network for potential matches among 77 participating transplant programs," said OPTN/UNOS president Charles Alexander, RN, M.S.N., M.B.A. United Network for Organ Sharing (UNOS) operates the OPTN under federal contract.

The donor recovery and transplant operations all took place Dec. 6. Mr. Crowder received a transplant at Barnes-Jewish Hospital in St. Louis and Ms. Niedzwiecki was transplanted at Dartmouth-Hitchcock Medical Center in Lebanon, N.H. Ms. Burkes donated her kidney at Barnes-Jewish and Ms. Richard underwent surgery at Dartmouth-Hitchcock.

<http://optn.transplant.hrsa.gov/resources/KPDPP.asp>



Exchange in practice

- In practice, the problem has a few twists...
 - US doctors think of compatibility as 0-1, which makes preferences different than the strict ranking in the housing model.
 - At first, doctors wanted to limit to pairwise trades, and rule out list exchange.
 - Compatible donors may not participate.
- A slightly simplified algorithm can be used.



Three-Way exchange

- It is possible but tricky to do multi-way exchanges, but they can help (esp. three-way).
- Pair is x - y if patient and donor have blood type x - y .
- Consider a population consisting of
 - O-B, O-A, A-B, A-B, B-A (blood type incompatible)
 - A-A, A-A, A-A, B-O (HLA incompatible)
- Assume there is no HLA problem across pairs
 - Two-way (A-B,B-A), (A-A,A-A), (O-B,B-O)
 - Three-way: (A-B,B-A), (A-A,A-A,A-A), (B-O,O-A,A-B).



Gains from Three-Way

- An odd number of A-A pairs can be transplanted.
- O-type donors can facilitate three transplants rather than two.
- In practice, O-type donors are short relative to demand, so useful to leverage them.
- Four-way exchanges also can help...



Donor Chains

- In July 2007, Alliance for paired donations started an “Altruistic Donor Chain”
- Altruistic donor in Michigan donated kidney to woman in Phoenix.
- Husband of Phoenix woman gave kidney to woman in Toledo.
- Her mom gave kidney to patient A in Columbus, whose daughter simultaneously gave kidney to patient B in Columbus.
- Now patient B’s sister is looking to donate....

Service to Community

Nation's longest kidney transplant chain reaches 34

by Tyler Greer

December 22, 2014 | [Print](#) | [Email](#)

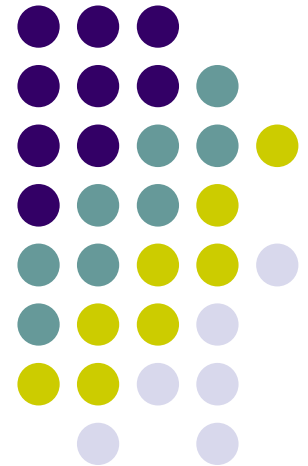
Tommy Thompson was born in December of 1973 with a horseshoe kidney, a condition in which the kidneys fuse together at the lower end during fetal development. It's a debilitating disorder that led to more than 40 surgeries of different types for Thompson by the time he was a young adult. Some of the surgeries were aimed at correcting the condition; others were an attempt to make dialysis possible so he could stay alive.

Finally, on Dec. 19 in **UAB Hospital** at the **University of Alabama at Birmingham**, at age 41, Thompson had the surgery he has needed most — a living-donor kidney transplant. Thompson received his kidney through UAB's record-breaking kidney chain. The chain, which began in December 2013, has matched 34 living donors with 34 recipients to create the longest kidney-transplant chain ever recorded in the United States; **previously, the longest chain on record** involved 30 donors and recipients in 17 hospitals around the country. All 34 recipients in the UAB kidney chain have been transplanted at UAB Hospital or **Children's of Alabama**.



"This will be my first time to be a healthy person. You don't know what that means. It's hard to describe what that means. I just know I'm looking forward to it, and I can't wait," said Thompson.

School Choice





Background

- Neighborhood schools
- *school choice* – flexibility for families, and competition between schools
- Today: designing school choice programs.



Design Objectives



- Efficient placements
- “Fair” procedure and outcomes
- Easy to understand and use



Approaches

- Abdulkadiroglu and Sonmez (2003, *AER*) - many placement mechanisms are flawed.
- Boston, New York, Chicago, San Francisco reformed their mechanisms.
- Active area of research
 - designing improved mechanisms,
 - studying the performance of mechanisms in use,
 - also, partially random nature of allocations has facilitated studies of school effectiveness in training students.

School Choice Model



- Set of students S and schools C
 - Each student can go to one school
 - Each school c can admit q_c students
 - Each student has strict preferences over schools.
 - Each school has a strict “priority order” over students.
- “Many-to-one” version of the marriage model.

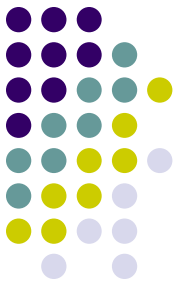


Matching

Matching μ is a function on the set of agents and schools such that

1. $\mu(s)$ is a school
2. $\mu(c)$ is a set of students
3. $s \in \mu(c) \leftrightarrow c = \mu(s)$
4. $|\mu(c)| \leq q_c$

Stability



Stability for school choice:

- **individual rationality** – no student wants to drop out and no school wants to kick out a student.
- **non-wastefulness** – no student can find a better school with an empty seat (school finds the student acceptable)
- **no blocking pair** – no student can find a better school which will displace someone to accept that student.

Stability as Fairness



- *Individual rationality* means no student can be forced to attend a school they don't want to attend, and no school can be forced to take a student they view as unqualified.
- *No blocking pair* means *no justified envy*. That is, there is no student s who gets a school they prefer less than c , only to see a student with lower priority end up at c .



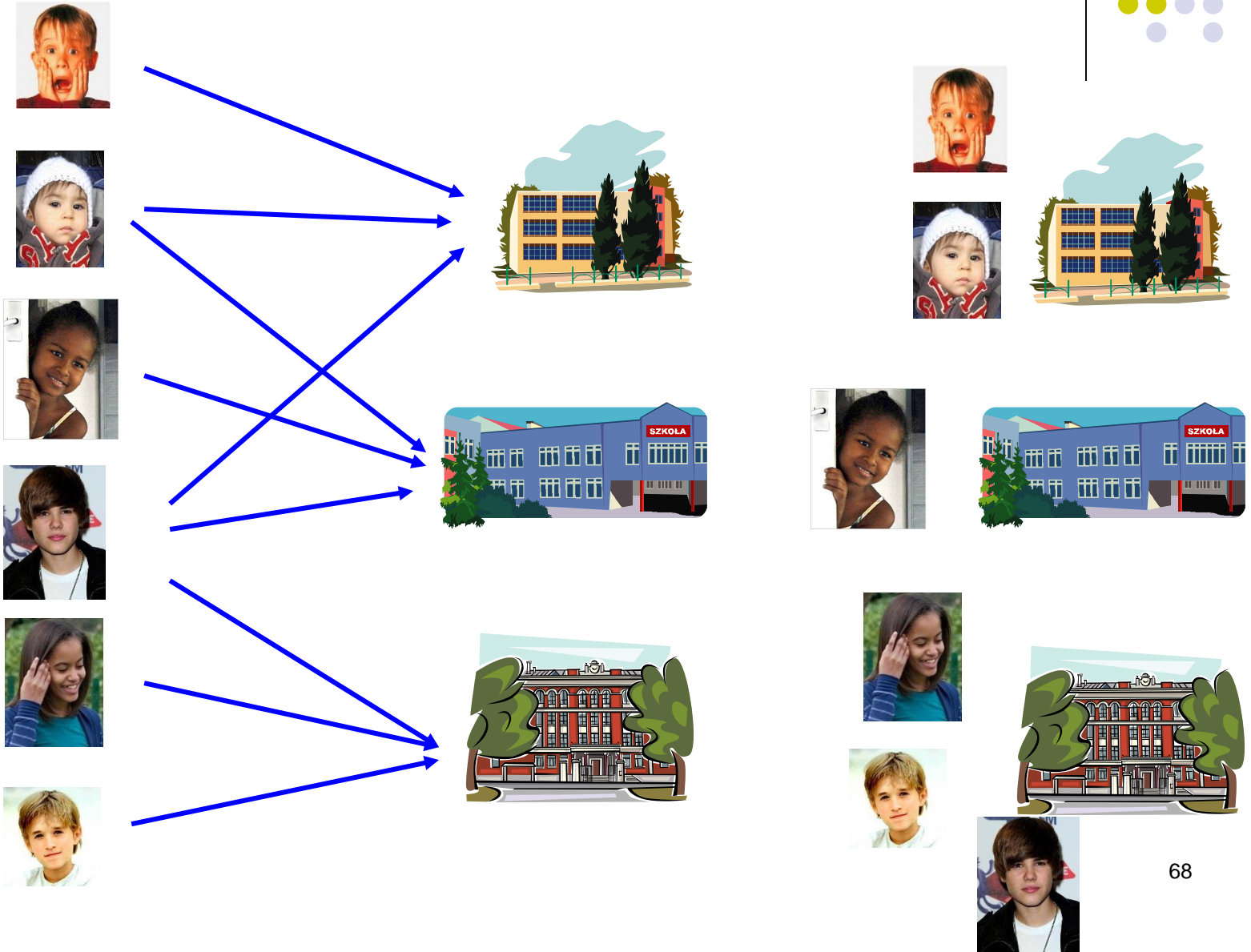
I. BOSTON MECHANISM (IMMEDIATE ACCEPTANCE)

- **Step 1:** Each school considers those students who listed it as her first choice. Those students with the highest priority for that school are permanently assigned to it. The rest are rejected. School quotas are updated.

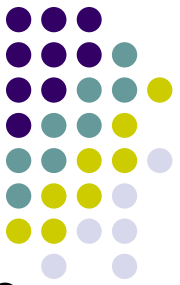
In general;







- **Step k , $k > 1$:** Each school with available seats considers those students who listed it as her **k -th** choice. Those students with the highest priority for that school are permanently assigned to it. The rest are rejected. School quotas are updated.
- **Note:** allocation at every step is final

Boston Mechanism in Pictures



Example: Gaming under Boston






1	2	3
		
A	B	B
.	.	A
.	.	C
		
A	B	C
3	2	.
1	3	.

Step 1

Step 2

Step 3

A	B	C
		
1	2, 3	
Full	Full	
		3



Problems with Boston

- **Boston mechanism is not strategy-proof.**
 - If you don't put your priority school high on your rank list, you may lose it!

Example: you want school A most and B second. You have high priority at B but not A. Both are in high demand so to get in to either, you need to rank it first *and* have high priority. It will be best to rank B first.
 - This was well-understood by Boston parents, and frequently showed up on parent message boards.

Quote from media (St. Petersburg Times)



“Make a realistic, informed selection on the school you list as your first choice. It’s the cleanest shot you will get at a school, but if you aim too high, you might miss.

Here is why: If the random computer selection rejects your first choice, your chances of getting your second choice is school are greatly diminished. That’s because you fall in line behind everyone who wanted your second choice as their first choice. You can fall even farther back in line as you get bumped down to your third, fourth and fifth choices.”



Problems with Boston

- **Boston mechanism is also unfair...**
 - Doesn't necessarily lead to stable outcomes.

Example: Consider the situation on the prior slide and suppose the family decides to take a risk, puts A first, and then ends up at much less preferred school C. The outcome is not stable because the family has high priority at B and prefers it to C.
 - Disadvantaged families that don't know how to game the system.



Boston in Practice

- Students in K,6,9 submit preferences
- Students have priorities as follows
 - Students already at a school.
 - Students in the walk zone and siblings at school
 - Students with siblings at school
 - Students in the walk zone
 - Everyone else
- Abdulkadiroglu et al. found that 19% listed two over-demanded schools as top two choices and about a quarter ended up unassigned – ugh.



- **Theorem:** Under complete info, the set of NE of Boston are equivalent to the stable set.
- **Theorem:** Under incomplete info, there are settings where the set of BNE of Boston dominate the DA outcome.



II. Deferred Acceptance

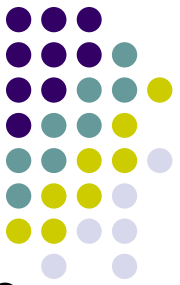
- What about the student proposing DA?
 - We know this leads to a stable match.
 - And the stable matching that is best for all the students, and they are the ones whose welfare we care about.
 - Plus it's strategy-proof for the students, and we may not be worried about schools if priorities are clearly stated.
- So our earlier results indicate that student-proposing DA has attractive properties...















But is it efficient?

- Stable matchings can be inefficient
- Consider students s_1, s_2, s_3 , and schools A, B
 - $P(s_1): B > A$
 - $P(s_2): A$
 - $P(s_3): A > B$
 - $P(A): s_1 > s_2 > s_3$
 - $P(B): s_3 > s_1$
 - Schools have one slot
- Student-proposing DA $\Rightarrow (s_1, A), s_2, (s_3, B)$
- But every student prefers: $(s_1, B), s_2, (s_3, A)$

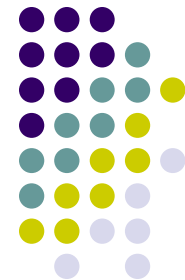
Deferred Acceptance: An example



1	2	3
		
A	B	B
		
C	A	C
		
2	1	3
3	2	2
1	3	1

	A	B	C
			
Step 1	1	2, 3	
Step 2	1, 3	2	
Step 3	3	2, 1	
Step 4	3	1	2

Stability and Efficiency



- A stable matching, even a student-optimal stable matching, may not be Pareto efficient if students can trade positions.
- Q1: If we want to find a matching that's Pareto efficient for students, how could we do it?
- Q2: if we find a matching that's Pareto efficient for students, will it necessarily be stable?



TTC for School Choice?

- Schools are different from houses because
 - A school has multiple positions not just one
 - A student can have priority at multiple schools.
- Still, we can adapt TTC to this setting
 - Abdulkadiroglu and Sonmez (2003) explain how.



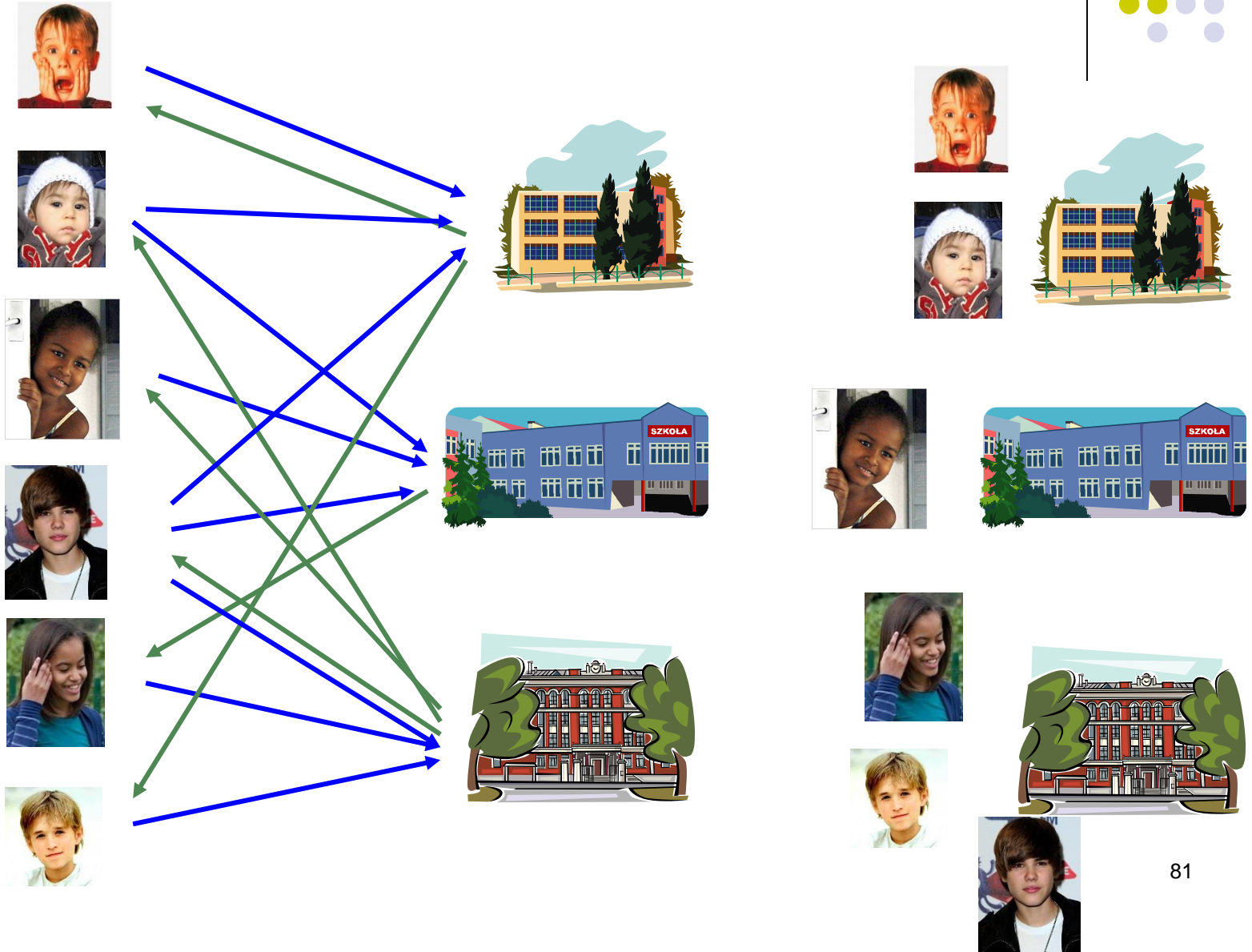
TTC for School Choice

- **TTC Algorithm**

- Each student points to its top-ranked school
- Each school points to its top-priority student
- Cycles are identified and removed: that is, any matched student is assigned and removed and the school to which she is assigned has its quota reduced by one seat.
- Point again and repeat the process...

- **TTC allows students to “trade” priorities.**

TTC for Schools in Pictures





Why TTC for Schools?

Theorem. The outcome of TTC is efficient and TTC is group strategy-proof.

- Proof is just like the housing problem
 - Strategy-proofness is the same.
 - Efficiency is just like proof of core outcomes.

The Random Assignment Problem



The random assignment problem



- Problem components
 - objects (e.g., houses, offices, tasks)
 - agents (e.g., tenants, staff, workers)
 - preferences (e.g. ROL's)
- Problem solutions
 - (deterministic/random) assignment mechanisms



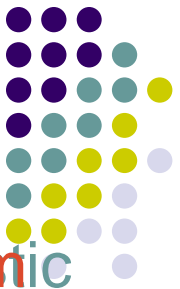
Example:

$N = \{1, 2, 3, 4\}, \quad H = \{a, b, c, d\}$

	R_1	R_2	R_3	R_4
1/	a	1/ a	1/ b	1/ b
2	b	2 b	2 a	2/ c
1/	c	1/2 d	1/ c	3/ d
3/	d	c	3/ d	6 a
6			6	

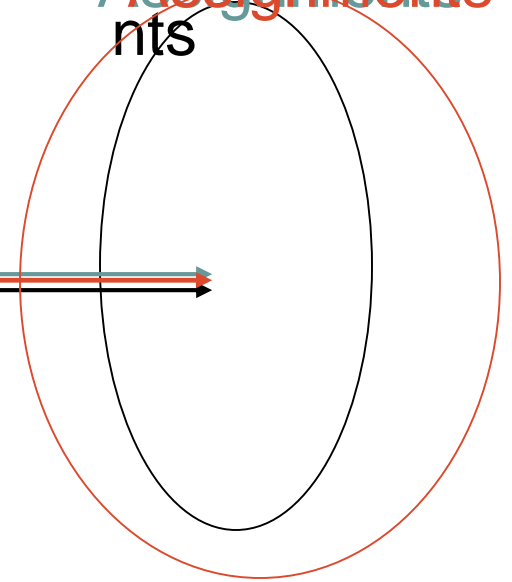
Not very fair!

Random assignment
more promise for fair



What is a mechanism?

(R)





- Model
- Random Priority (RP)
 - An equivalence (Abdulkadiroglu & Sonmez, 1998)
- A mechanism from recent theory: Probabilistic Serial (PS)
- Discussion: What's the matter with RP and TTC?
- Two new mechanisms
 - Top trading cycles from equal division
 - Random Priority [*k]
- An equivalence theorem

The Model



- Agents $N=\{1, 2, \dots, n\}$
- Objects (Houses) $H=\{h_1, h_2, \dots, h_n\}$
- A list of strict preferences $R=(R_i)_{i \in N}$
- An **assignment** is a bijection $\mu:N \rightarrow H$
- A **random assignment** is a *bistochastic* matrix $P=[p_{ix}]_{i \in N, x \in H}$ where

$$p_{ix} := \text{Prob}\{\text{agent } i \text{ gets object } x\}$$

Fact: A random assignment can be expressed as a lottery over deterministic assignments and vice versa.

- A **mechanism** is a function ϕ s.t. $\phi(R)$ is the associated random assignment for the problem R

Random Priority



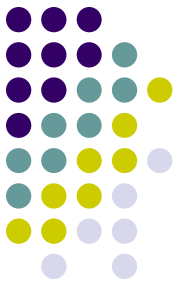
RP Algorithm

1. Draw a random ordering of agents from the uniform distribution.
2. Compute the associated *serial dictatorship*

Formally, $RP(R) = 1/n! \sum_f SD^f(R)$

i.e., average of $n!$ serial dictatorships

An interesting equivalence



Theorem: Random serial dictatorship is equivalent to the following mechanism:

Core from Random Endowments

1. Randomly assign each object to a different agent
2. Compute the associated **core** outcome
(via the **Top Trading Cycles** procedure)

Welfare notions:



- **Ex post-efficiency:** A random assignment is **ex-post efficient** iff all lotteries that induce it have support over Pareto efficient assignments.
- **Ordinal efficiency:** A random assignment is **ordinally efficient** iff it is not stochastically dominated by another random assignment.

A random matching **P** **ordinally dominates** **Q** iff for all **i** and **s**:

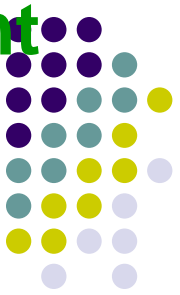
Prob {student **i** is assigned to **s** or a better school under **P**} \geq

Prob {student **i** is assigned to **s** or a better school under **Q**};

holding strictly for some student **j**.

Surprise: Random Priority is *not* fully efficient !!!

(Bogomolnaia & Moulin, 2001)



Example: Suppose $R_1 = R_2: a b c d$
 $R_3 = R_4: b a d c$

RP
Assignment

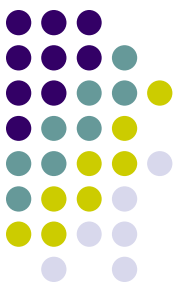
	a	b	c	d
1	5/12	1/12	5/12	1/12
2	5/12	1/12	5/12	1/12
3	1/12	5/12	1/12	5/12
4	1/12	5/12	1/12	5/12

Alternative
Assignment

	a	b	c	d
1	1/2	0	1/2	0
2	1/2	0	1/2	0
3	0	1/2	0	1/2
4	0	1/2	0	1/2

The Challenger: Probabilistic Serial

(Bogomolnaia & Moulin, JET 2001)



PS Algorithm

Think of each object as an infinitely divisible good. If an agent eats a p_{ix} of object x during the procedure, then PS allocates him object x with probability p_{ix} .

Step 1: Each agent eats away from his favorite object at the same speed.
Stop when an object is completely exhausted.

In general,

Step k , $k \geq 2$: Consider the remaining objects with the remaining units of them. Each agent eats away from his favorite object at the same speed. Stop when an object is completely exhausted.

Theorem (Bogomolnaia & Moulin): PS finds the central point of the ordinally efficient set.