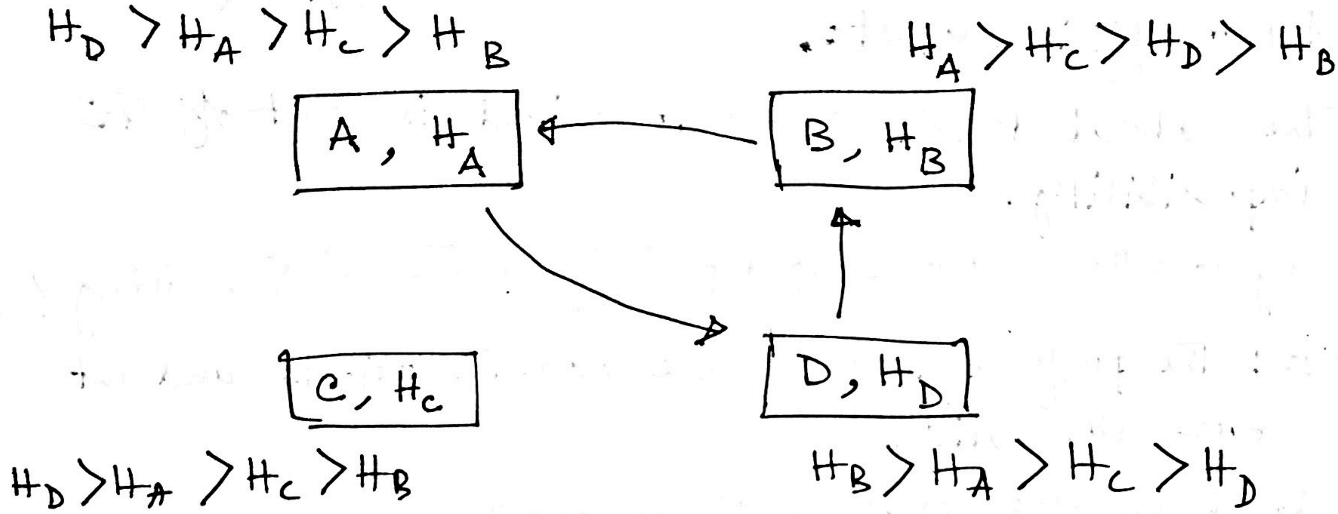
One-sided Matching

Why one sided? Only one side of the market has preference over the other side. The other side ~~is~~ consists of objects/resources, etc., that does not typically have a preference over the ~~other~~ first side.

Example: House allocation, each agent comes to the market with "initial endowments" (Shapley & Scarf 74 model)



Money is disallowed. Only option is to exchange/reallocation. Is there a better allocation? Yes. If the agents get the houses of the agent they point to in the figure.
The new allocation is Pareto better than the initial endowments.

Formal model of one-sided matching problem.

Set of objects $M = \{a_1, a_2, \dots, a_m\}$

Set of agents $N = \{1, 2, \dots, n\}$, $m \geq n$.

- Objects can be houses, jobs, projects, etc.
- Each agent has a linear ~~or~~ order over the objects

$R_i \rightarrow$ complete : $\forall a, b \in M$ either $a R_i b$ or $b R_i a$.

transitive : if $a R_i b$ and $b R_i c \Rightarrow a R_i c$

anti-symmetric : if $a R_i b$ and $b R_i a \Rightarrow a = b$

We will denote such linear orders with P_i

$P = (P_1, P_2, \ldots, P_n)$ is a preference profile

- $\mathcal{M}$: set of all possible linear orders over M.
- $P_i(k, S)$ is the k-th top object in $S \subseteq M$ in P_i.

Questions we are interested in is a collective decision problem that satisfies strategyproofness, Pareto efficiency, etc.

The natural result comes to mind is that of GS impossibility.

* Why does the current setup depart from the G-S setting?

Note: The preferences here are over the objects and not over alternatives

What is an alternative in this context?

A matching / assignment of the objects to the agents.

A <u>feasible</u> matching is a mapping

$a : N \to M$ injective (one-to-one)

if $a(i) = a(j) \Rightarrow i = j$

- each agent gets exactly one item and every item goes to at most one agent.
- Set of alternatives $A = \{ a : N \to M ;\ \text{injective} \}$

Now, consider two alternatives a and a' where agent i gets the same item. There is no preference profile where either $a \, P_i \, a'$ or $a' \, P_i \, a$. The agent is always indifferent between a and a'. Hence, the domain of single object allocation ~~viol~~ violates the unrestricted assumption of G-S theorem.

Hence G-S theorem does not hold here.

So, we can expect to have non-trivial truthful and efficient mechanisms.

Before beginning non-trivial mechanisms, consider a simple (truthful) mechanism.

$\boxed{\text{Ex:}}$ SCF $f : \mathcal{M}^n \to A$. Define a fixed priority (serial dictatorship) mechanism.

A priority is an ordering over the agents order $(\sigma(1), \sigma(2) \cdots$

$\sigma : N \to N$, bijection (permutation function)

The mechanism: Every agent in the order σ picks her favorite from the leftover list.

$$a(\sigma(1)) = P_{\sigma(1)}(1, M)$$

$$a(\sigma(2)) = P_{\sigma(2)}(1, M \setminus \{a(\sigma(1))\})$$

$$a(\sigma(3)) = P_{\sigma(3)}(1, M \setminus \{a(\sigma(1)), a(\sigma(2))\})$$

$$\vdots$$

construct the allocation for a given P and priority σ.

$$\underline{f^{\sigma}(P) = a}$$

- It is a generalization of dictatorship
- Easy to see it is strategyproof.

Defn: (Strategyproofness)

An SCF $f: \mathcal{M}^n \to A$ is strategyproof if

$$f_i(P_i, P_{-i}) \; P_i \; f_i(P_i', P_{-i}) \quad \forall P_i, P_{-i}, P_i'$$

$$\text{or} \quad f_i(P_i, P_{-i}) = f_i(P_i', P_{-i}) \quad \forall i \in N$$

Another desirable property is "Efficiency".

Defn: An SCF f is efficient if for every preference profile, there exists no matching $a \neq f(P)$ s.t. $a(i) = f_i(P)$ or $a_i \, P_i \, f_i(P)$.

Is serial dictatorship efficient? Yes.

How to prove? Suppose f^σ is not efficient. Then $\exists P$ s.t. $\exists a \neq f^\sigma(P)$ satisfying $a(i) \, P_i \, f_i^\sigma(P)$ or $a(i) = f_i^\sigma(P)$, ~~$\forall i \in N$~~ i.e., either agent i gets the same house or gets a better house, $\forall i \in N$.

Say, the first agent j in the priority order of σ that has $a(j) \, P_j \, f_j^\sigma(P)$. But it means that $a(j)$ was available when j's turn came to pick. According to the serial dictatorship, then agent j can't pick $f_j^\sigma(P)$ which is less preferred than $a(j)$.

The proof of strategyproofness is also easy.

The agents before i in the order σ picks houses and i can't influence it. Agent i gets the best house from the remaining houses, hence there is no reason to misreport.

However, serial dictatorship is not the only mechanism in this domain.

$\boxed{\text{Example :}}$ $N = \{1, 2, 3\}$, $M = \{h_1, h_2, h_3\}$

The SCF is same as serial dictatorship, but the priority order changes based on the preference of agent 1 in the following way.

$$\sigma = \begin{cases} (1, 2, 3) & \text{if } P_1(1) = h_1 \\ (2, 1, 3) & \text{if } P_1(1) \neq h_1 \end{cases}$$

Is this strategyproof? Yes

Players 2 and 3 can't change the priority order, hence they can't manipulate. Player 1 can change it, but in case 1: $P_1(1) = h_1$, it gets her most favorite house. In case 2: $P_1(1) \neq h_1$, if she misreports to $P_1(1) = h_1$, she will be assigned h_1 , which she prefers less than her top choice, say $b \in \{h_2, h_3\}$

Since agent 1 picks second if she reports truthfully, she can either get h_1 or better (if agent 2's preference is s.t.. $P_2(1) \neq b$). Agent 1 can get h_1 only if $P_2(1) = b$ and $P_1(2) = h_1$)

Is this efficient? Yes.

Every fixed priority serial dictatorship is efficient. The same argument as before applies.

This modification of the serial dictatorship also satisfies the two desired properties.

1-6

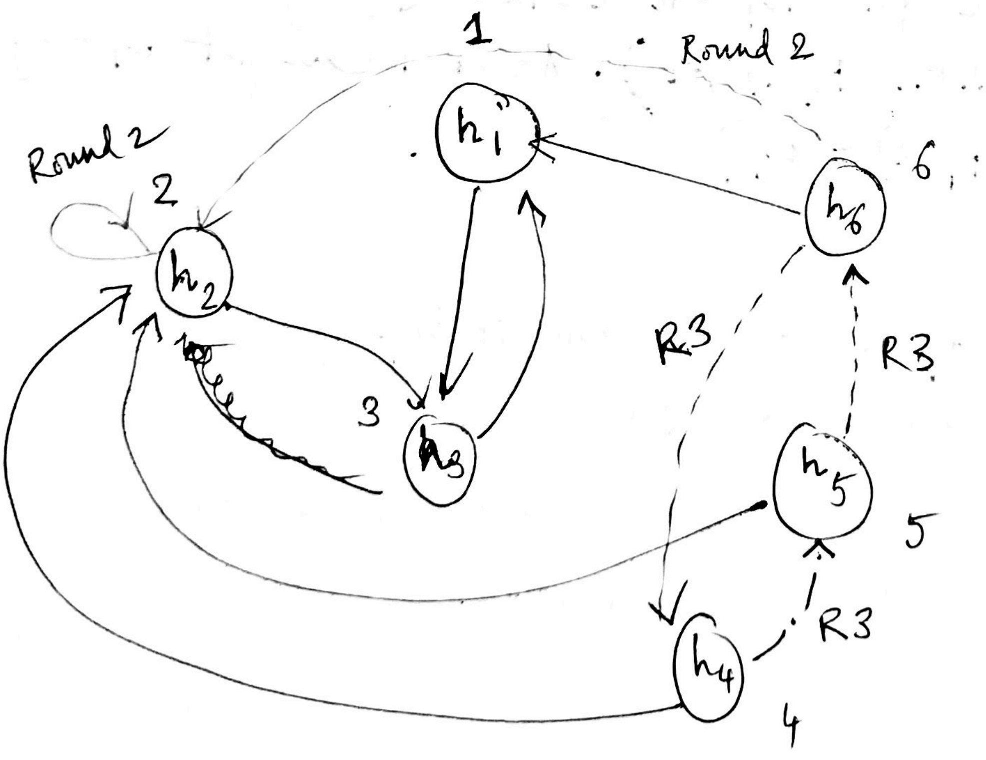
There is another (class of) mechanism(s) that satisfies a few desirable properties.

Top Trading Cycle with fixed endowments:

Illustration with example. Assume $|M| = |N| = 6$ (for simplicity)

Initialization: Each agent is endowed with some houses. Say, agents 1 to 6 have houses h_1 to h_6 respectively as endowments.

P_1	P_2	P_3	P_4	P_5	P_6
h_3	h_3	h_1	h_2	h_2	h_1
h_1	h_2	h_4	h_1	h_1	h_3
h_2	h_1	h_3	h_5	h_6	h_2
⋮	⋮	⋮	⋮	⋮	h_4



R1 : $1 \to h_3$
$3 \to h_1$

leaves

preference graph updated

R2 : $2 \to h_2$

leaves

R3 $4 \to h_5$
$5 \to h_6$
$6 \to h_4$

leaves

Algorithm terminates.