Reinforcement Learning China Summer School

RLChina 2021

Game Theory: Advanced Topics

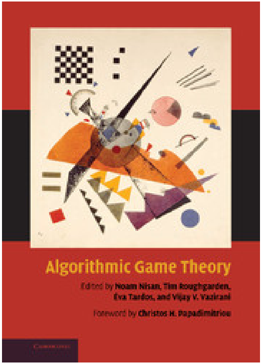
Zhengyang Liu

School of Computer Science
Beijing Institute of Technology

August 16, 2021

Table of Contents

Main References



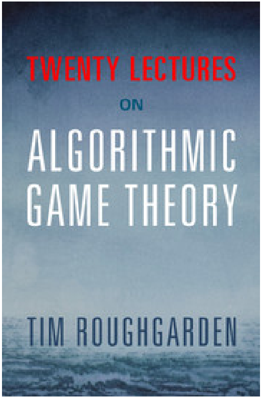


Table of Contents

Our Goal in MD

Goal

Understand how to design systems with strategic participants that have good performance guarantees.

Single-Item Auctions

Single-Item Auctions

- A seller, n (strategic) bidders and one item
- Each bidder i has a private value $v_i \geq 0$.
- Quaslinear utility: $u_i = v_i - p$ if she wins at price p, $u_i = 0$ o.w.

Single-Item Auctions

- A seller, n (strategic) bidders and one item
- Each bidder i has a private value $v_i \geq 0$.
- Quaslinear utility: $u_i = v_i - p$ if she wins at price p, $u_i = 0$ o.w.

Sealed-Bid Auctions

1. Each bidder i privately sends a bid b_i to the seller.
2. The seller decides the winner.
3. The seller decides the selling price.

Single-Item Auctions

- A seller, n (strategic) bidders and one item
- Each bidder i has a private value $v_i \geq 0$.
- Quaslinear utility: $u_i = v_i - p$ if she wins at price p, $u_i = 0$ o.w.

Sealed-Bid Auctions

1. Each bidder i privately sends a bid b_i to the seller.
2. The seller decides the winner. Easy in common.
3. The seller decides the selling price.

Single-Item Auctions

- A seller, n (strategic) bidders and one item
- Each bidder i has a private value $v_i \geq 0$.
- Quaslinear utility: $u_i = v_i - p$ if she wins at price p, $u_i = 0$ o.w.

Sealed-Bid Auctions

1. Each bidder i privately sends a bid b_i to the seller.
2. The seller decides the winner. Easy in common.
3. The seller decides the selling price. Many different ways...

First-Price Auctions

- The winner pays her bid.
- Figure out how to bid?
- Predict what will happen?

First-Price Auctions

- The winner pays her bid.
- Figure out how to bid?
- Predict what will happen?

- Your value is sum of your birth month and the day of your birth, i.e. from 2 (Jan 1) to 43 (Dec 31).
- The other bidder whose value is constructed with the same way.
- How do you bid to max your expected utility?
- What if there are one more bidder in the auction?

Second-Price Auctions

- The winner pays the second-highest bid.
- Truthfully bidding is a dominant strategy, i.e., guarantee to max her utility, no matter what other bidders do.

Second-Price Auctions

- The winner pays the second-highest bid.
- Truthfully bidding is a dominant strategy, i.e., guarantee to max her utility, no matter what other bidders do.
- Fix a bidder i, and her value is v_i, other bids $\mathbf{b}_{-i}$. Let $B = \max_{j \neq i} b_j$, we consider two cases:
- If $v_i < B$, easy.
- If $v_i \geq B$, the maximum utility she can obtain is $\max\{0, v_i - B\} = v_i - B$.

Second-Price Auctions

- The winner pays the second-highest bid.
- Truthfully bidding is a dominant strategy, i.e., guarantee to max her utility, no matter what other bidders do.
- Fix a bidder i, and her value is v_i, other bids $\mathbf{b}_{-i}$. Let $B = \max_{j \neq i} b_j$, we consider two cases:
- If $v_i < B$, easy.
- If $v_i \geq B$, the maximum utility she can obtain is $\max\{0, v_i - B\} = v_i - B$.
- Another property, individual rationality, i.e. always get non-negative utility.

Ideal Auctions

Dominant-Strategy Incentive Compatible (DSIC)

An auction is *DSIC* if bidding truthfully is always a dominant strategy for every bidder and it can help them get non-negative utility.

Ideal Auctions

Dominant-Strategy Incentive Compatible (DSIC)

An auction is *DSIC* if bidding truthfully is always a dominant strategy for every bidder and it can help them get non-negative utility.

Theorem [Second-Price Auctions are Ideal]

1. It is DSIC.
2. Welfare maximization: $\sum_{i=1}^{n} v_i x_i$, where x_1 is 1 if bidder i wins, 0 o.w.
3. It is computationally efficient.

Multi-Item Auctions

Sponsored Search Auctions

- Assume that we have k "slot" for some search web page with AD keywords, e.g., phone, car.

Sponsored Search Auctions

- Assume that we have k "slot" for some search web page with AD keywords, e.g., phone, car.
- Click-through rates (CTR) make the slots different, say $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$.

Sponsored Search Auctions

- Assume that we have k "slot" for some search web page with AD keywords, e.g., phone, car.
- Click-through rates (CTR) make the slots different, say $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$.
- Each advertiser is only interested in the click on her link, the private value is v_i. So the expected value from slot j is $v_i\alpha_j$.

Sponsored Search Auctions

- Assume that we have k "slot" for some search web page with AD keywords, e.g., phone, car.
- Click-through rates (CTR) make the slots different, say $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$.
- Each advertiser is only interested in the click on her link, the private value is v_i. So the expected value from slot j is $v_i\alpha_j$.
- Two things: allocations and payments.

Sponsored Search Auctions

- Assume that we have k "slot" for some search web page with AD keywords, e.g., phone, car.
- Click-through rates (CTR) make the slots different, say $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$.
- Each advertiser is only interested in the click on her link, the private value is v_i. So the expected value from slot j is $v_i \alpha_j$.
- Two things: allocations and payments.
- Given truthful bidding, greedy allocation is optimal.

Sponsored Search Auctions

- Assume that we have k "slot" for some search web page with AD keywords, e.g., phone, car.
- Click-through rates (CTR) make the slots different, say $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$.
- Each advertiser is only interested in the click on her link, the private value is v_i. So the expected value from slot j is $v_i\alpha_j$.
- Two things: allocations and payments.
- Given truthful bidding, greedy allocation is optimal.
- Say bids are $b_1 \geq \cdots \geq b_n$. The price of bidder i is

$$p_i(\mathbf{b}) = \sum_{j=i}^{k} b_{j+1} \frac{\alpha_j - \alpha_{j+1}}{\alpha_i}.$$

(By Myerson's Lemma)

- A famous simplified version is the generalized second-price (GSP) auction.
- Allocation: assign the ith highest bidder to the ith best slot.
- Payment: charge the bidder i the $(i+1)$th highest bid per click.

- A famous simplified version is the generalized second-price (GSP) auction.
- Allocation: assign the ith highest bidder to the ith best slot.
- Payment: charge the bidder i the $(i+1)$th highest bid per click.

Non-truthfulness

- A famous simplified version is the generalized second-price (GSP) auction.
- Allocation: assign the ith highest bidder to the ith best slot.
- Payment: charge the bidder i the $(i+1)$th highest bid per click.

Non-truthfulness

- $\alpha_1 = 1, \alpha_2 = 0.4$ and $v_1 = 7, v_2 = 6, v_3 = 1$.

- A famous simplified version is the generalized second-price (GSP) auction.
- Allocation: assign the ith highest bidder to the ith best slot.
- Payment: charge the bidder i the $(i+1)$th highest bid per click.

Non-truthfulness

- $\alpha_1 = 1, \alpha_2 = 0.4$ and $v_1 = 7, v_2 = 6, v_3 = 1$.
- If bidding truthfully, $u_1 = \alpha_1(v_1 - p_1) = 1 \cdot (7 - 6) = 1$.

- A famous simplified version is the generalized second-price (GSP) auction.
- Allocation: assign the ith highest bidder to the ith best slot.
- Payment: charge the bidder i the $(i+1)$th highest bid per click.

Non-truthfulness

- $\alpha_1 = 1, \alpha_2 = 0.4$ and $v_1 = 7, v_2 = 6, v_3 = 1$.
- If bidding truthfully, $u_1 = \alpha_1(v_1 - p_1) = 1 \cdot (7 - 6) = 1$.
- If setting $b_1' = 5$, then $u_1' = \alpha_2(v_1 - p_1') = 0.4 \cdot (7 - 1) = 2.4$.

Combinatorial Auctions

- n bidders with a set M of m items.
- Each bidder i has a value function $v_i : 2^M \to \mathbb{R}_{\geq 0}$.
- The outcome is $\mathbf{S} = (S_1, \ldots, S_n)$, where $\cup_i S_i \subseteq M$ and S_i's are disjoint each other.

VCG Mechanism

- Every general environment admits a DSIC welfare-maximizing mechanism.

VCG Mechanism

- Every general environment admits a DSIC welfare-maximizing mechanism.
- Given $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, the allocation rule $\mathbf{x}$ is

$$\mathbf{x}(\mathbf{b}) = \mathrm{argmax}_{\mathbf{S}} \sum_{i=1}^{n} b_i(\mathbf{S}).$$

VCG Mechanism

- Every general environment admits a DSIC welfare-maximizing mechanism.
- Given $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, the allocation rule $\mathbf{x}$ is

$$\mathbf{x}(\mathbf{b}) = \mathrm{argmax}_{\mathbf{S}} \sum_{i=1}^{n} b_i(\mathbf{S}).$$

- Charge an agent her externality,

$$p_i(\mathbf{b}) = \underbrace{\left(\max_{\mathbf{S}} \sum_{j \neq i} b_j(\mathbf{S}) \right)}_{\text{without } i} - \underbrace{\sum_{j \neq i} b_j(\mathbf{S}^*)}_{\text{with } i},$$

where $\mathbf{S}^* = \mathbf{x}(\mathbf{b})$.

Table of Contents

Preliminaries

- We focus on two-player games, a.k.a, bimatrix games ($A, B \in [0,1]^{n\times n}$).
- Mixed strategies: $x, y \in \Delta^n$, where $\Delta^n := \left\{z \in \mathbb{R}^n_{\geq 0} \mid \sum_{i=1}^n z_i = 1\right\}$.
- Nash equilibrium (x, y): if $x^T A y \geq x'^T A y$ and $x^T B y \geq x^T B y'$.

Preliminaries

- We focus on two-player games, a.k.a, bimatrix games ($A, B \in [0,1]^{n\times n}$).
- Mixed strategies: $x, y \in \Delta^n$, where $\Delta^n := \left\{z \in \mathbb{R}^n_{\geq 0} \mid \sum_{i=1}^n z_i = 1\right\}$.
- Nash equilibrium (x, y): if $x^T Ay \geq x'^T Ay$ and $x^T By \geq x^T By'$.
- Our goal: argue that finding an NE is intractable.

Complexity 101

Complexity 101

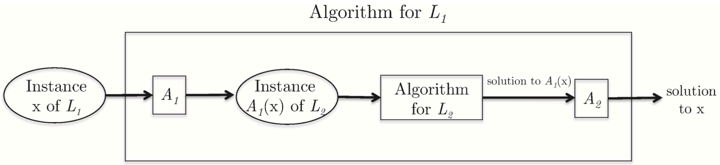
- NP (Non-deterministic Polynomial): homework vs grading

Complexity 101

- NP (Non-deterministic Polynomial): homework vs grading
- F(unctional) NP: Search vs decision

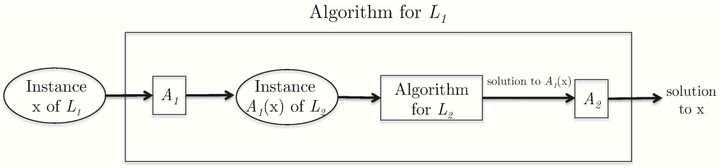
Complexity 101

- NP (Non-deterministic Polynomial): homework vs grading
- F(unctional) NP: Search vs decision
- Reduction from L_1 to L_2:



Complexity 101

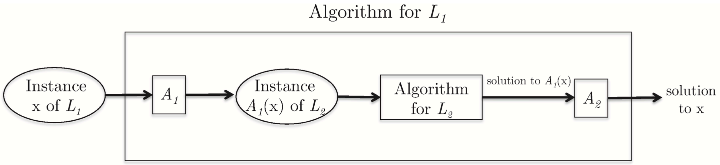
- NP (Non-deterministic Polynomial): homework vs grading
- F(unctional) NP: Search vs decision
- Reduction from L_1 to L_2:



- T(otal) FNP: every instance has at least one solution.

Complexity 101

- NP (Non-deterministic Polynomial): homework vs grading
- F(unctional) NP: Search vs decision
- Reduction from L_1 to L_2:



- T(otal) FNP: every instance has at least one solution.
- PPAD $\subseteq$ TFNP

PPAD

Problem $\textsc{End-of-A-Line}$

Input: two boolean circuits $S, P : \{0,1\}^n \to \{0,1\}^n$, such that $P(0^n) = 0^n \neq S(0^n)$.

Output: $x \in \{0,1\}^n$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$.

PPAD

Problem End-of-A-Line

Input: two boolean circuits $S, P : \{0,1\}^n \to \{0,1\}^n$, such that $P(0^n) = 0^n \neq S(0^n)$.
Output: $x \in \{0,1\}^n$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$.

- **P**olynomial **P**arity **A**rgument on a **D**irected graph

PPAD

Problem END-OF-A-LINE

Input: two boolean circuits $S, P : \{0,1\}^n \to \{0,1\}^n$, such that $P(0^n) = 0^n \neq S(0^n)$.
Output: $x \in \{0,1\}^n$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$.

- **P**olynomial **P**arity **A**rgument on a **D**irected graph
- Problem X is in PPAD if there is a reduction from X to END-OF-A-LINE.

PPAD

Problem END-OF-A-LINE

Input: two boolean circuits $S, P : \{0,1\}^n \to \{0,1\}^n$, such that $P(0^n) = 0^n \neq S(0^n)$.
Output: $x \in \{0,1\}^n$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$.

- **P**olynomial **P**arity **A**rgument on a **D**irected graph
- Problem X is in PPAD if there is a reduction from X to END-OF-A-LINE.
- X is PPAD-hard if there is a reversed reduction.

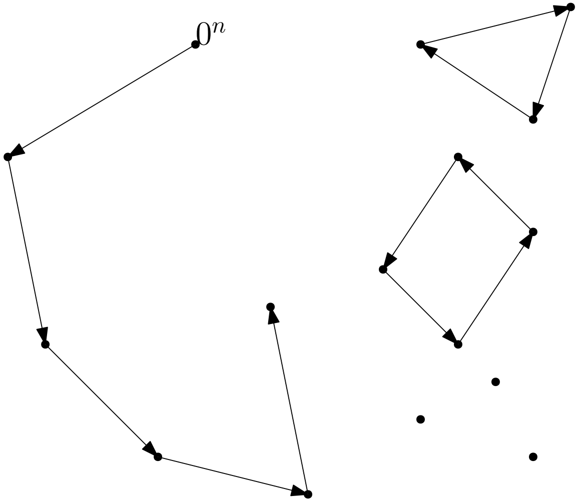


Figure: Sketch of End-of-A-Line

Proof Overview — Reductions![1]

1. End-of-A-Line to 2D-End-of-A-Line
2. to Discrete Fixed-Point
3. Finding a fixed point with Boolean Circuits
4. Simulating gates with game gadgets
5. Putting all gates together

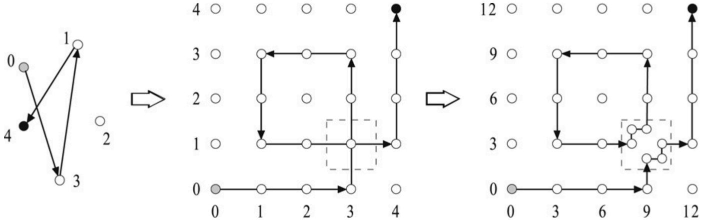
[1] Chen, X. and Deng, X., 2007. Recent development in computational complexity characterization of Nash equilibrium. Computer Science Review, 1(2), pp.88-99.

Proof Overview — Reductions![1]

1. END-OF-A-LINE to 2D-END-OF-A-LINE
2. to Discrete Fixed-Point
3. Finding a fixed point with Boolean Circuits
4. Simulating gates with game gadgets
5. Putting all gates together

[1] Chen, X. and Deng, X., 2007. Recent development in computational complexity characterization of Nash equilibrium. Computer Science Review, 1(2), pp.88-99.

Embedding a Path on a 2D Space



Generalized Circuits

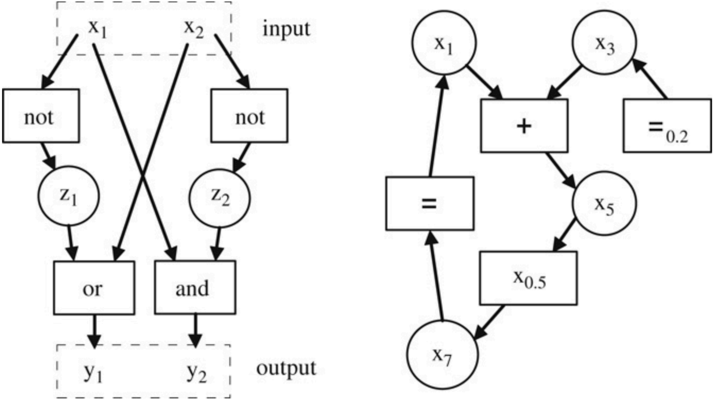


Figure: Two differences: 1. type of gates 2. has cycle

An Addition Gate Gadget

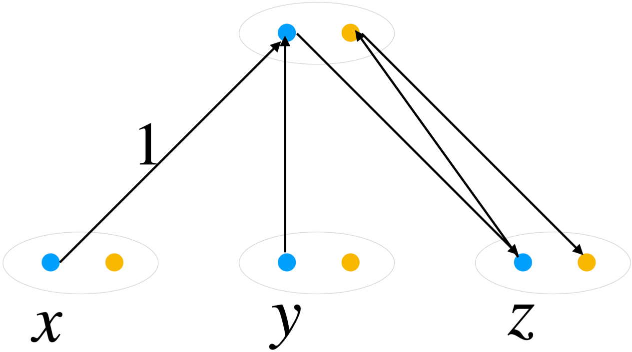


Figure: We make sure each player acts as an output at most once.

All Gates

$\mathbf{L}[T]$ **and** $\mathbf{R}[T]$**, where gate** $T = (G, v_1, v_2, v, \alpha)$
Set $\mathbf{L}[T] = (L_{i,j}) = \mathbf{R}[T] = (R_{i,j}) = 0$, $k = \mathcal{C}(v)$, $k_1 = \mathcal{C}(v_1)$ and $k_2 = \mathcal{C}(v_2)$
$G_\zeta :\quad L_{2k-1,2k} = L_{2k,2k-1} = R_{2k-1,2k-1} = 1,\ R_{i,2k} = \alpha, \forall\, i : 1 \leq i \leq 2K.$
$G_{\times\zeta} : L_{2k-1,2k-1} = L_{2k,2k} = R_{2k-1,2k} = 1,\ R_{2k_1-1,2k-1} = \alpha.$
$G_= :\quad L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k-1,2k} = 1.$
$G_+ :\quad L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k-1} = R_{2k-1,2k} = 1.$
$G_- :\quad L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k} = R_{2k-1,2k} = 1.$
$G_< :\quad L_{2k-1,2k} = L_{2k,2k-1} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k} = 1.$
$G_\vee :\quad L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k-1} = 1,\ R_{i,2k} = 1/(2K),\ \forall\, i \in [2K].$
$G_\wedge :\quad L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k-1} = 1,\ R_{i,2k} = 3/(2K),\ \forall\, i \in [2K].$
$G_\neg :\quad L_{2k-1,2k} = L_{2k,2k-1} = R_{2k_1-1,2k-1} = R_{2k_1,2k} = 1.$

The Final Step

Generalized Matching Pennies: $G^* = (A^*, B^*)$, where

$$A^* = \begin{pmatrix} M & M & 0 & 0 & \cdots & 0 & 0 \\ M & M & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & M & M & \cdots & 0 & 0 \\ 0 & 0 & M & M & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & M & M \\ 0 & 0 & 0 & 0 & \cdots & M & M \end{pmatrix},$$

and $B^* = -A^*$.

The Final Step

Generalized Matching Pennies: $G^* = (A^*, B^*)$, where

$$A^* = \begin{pmatrix} M & M & 0 & 0 & \cdots & 0 & 0 \\ M & M & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & M & M & \cdots & 0 & 0 \\ 0 & 0 & M & M & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & M & M \\ 0 & 0 & 0 & 0 & \cdots & M & M \end{pmatrix},$$

and $B^* = -A^*$. Note that we do have other constructions:)

The Final Step

Generalized Matching Pennies: $G^* = (A^*, B^*)$, where

$$A^* = \begin{pmatrix} M & M & 0 & 0 & \cdots & 0 & 0 \\ M & M & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & M & M & \cdots & 0 & 0 \\ 0 & 0 & M & M & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & M & M \\ 0 & 0 & 0 & 0 & \cdots & M & M \end{pmatrix},$$

and $B^* = -A^*$. Note that we do have other constructions:)

Property

For every game $G = (A, B)$ with all entries in $A - A^*, B - B^*$ are small, every approximate NE (x, y) of G are almost "pairwise" uniform.

1. END-OF-A-LINE to 2D-END-OF-A-LINE
2. to Discrete Fixed-Point
3. Finding a fixed point with Boolean Circuits
4. Simulating gates with game gadgets
5. Putting all gates together

Table of Contents

Take-home Message

Take-home Message

- A *mechanism* is a method of mapping agents' reported valuations to an outcome.

Take-home Message

- A *mechanism* is a method of mapping agents' reported valuations to an outcome.
- Individual Rationality, Incentive Compatibility and Allocative Efficiency.

Take-home Message

- A *mechanism* is a method of mapping agents' reported valuations to an outcome.
- Individual Rationality, Incentive Compatibility and Allocative Efficiency.
- VCG mechanism.

Take-home Message

- A *mechanism* is a method of mapping agents' reported valuations to an outcome.
- Individual Rationality, Incentive Compatibility and Allocative Efficiency.
- VCG mechanism.
- General Games are (PPAD-)hard to solve. But we have some approximation algorithms.

What we didn't cover

What we didn't cover

- Myerson's Lemma

What we didn't cover

- Myerson's Lemma
- MD without Money, e.g., kidney exchange, stable matching...

What we didn't cover

- Myerson's Lemma
- MD without Money, e.g., kidney exchange, stable matching...
- Market Equilibria, PPA etc.

Thanks for listening!

RLChina 2021

https://lozycs.github.io/