

# **Mechanism design: Single-parameter environments**

**Alexandros Voudouris**

University of Oxford

# Single-item auctions

- A seller with **one item for sale**
- $n$  **agents**

# Single-item auctions

- A seller with **one item for sale**
- $n$  **agents**
- Each agent  $i$  has a **private value**  $v_i$  for the item
  - This value represents the **willingness-to-pay** of the agent; that is,  $v_i$  is the maximum amount of money that agent  $i$  is willing to pay in order to buy the item

# Single-item auctions

- A seller with **one item for sale**
- $n$  **agents**
- Each agent  $i$  has a **private value**  $v_i$  for the item
  - This value represents the **willingness-to-pay** of the agent; that is,  $v_i$  is the maximum amount of money that agent  $i$  is willing to pay in order to buy the item
- The utility of each agent is **quasilinear in money**:
  - If agent  $i$  loses the item, then her utility is 0
  - If agent  $i$  wins the item at price  $p$ , then her utility is  $v_i - p$

# Single-item auctions

- General structure of an auction:
  - **Input:** every agent  $i$  submits a bid  $b_i$  (*agents = bidders*)
  - **Allocation rule:** decide the winner
  - **Payment rule:** decide a selling price

# Single-item auctions

- General structure of an auction:
  - **Input:** every agent  $i$  submits a bid  $b_i$  (*agents = bidders*)
  - **Allocation rule:** decide the winner
  - **Payment rule:** decide a selling price
- Deciding the winner is easy: the highest bidder

# Single-item auctions

- General structure of an auction:
  - **Input:** every agent  $i$  submits a bid  $b_i$  (*agents = bidders*)
  - **Allocation rule:** decide the winner
  - **Payment rule:** decide a selling price
- Deciding the winner is easy: the highest bidder
- Deciding the selling price is more complicated

# Single-item auctions

- General structure of an auction:
  - **Input:** every agent  $i$  submits a bid  $b_i$  (*agents = bidders*)
  - **Allocation rule:** decide the winner
  - **Payment rule:** decide a selling price
- Deciding the winner is easy: the highest bidder
- Deciding the selling price is more complicated
  - A selling price of 0, creates a competition among the bidders as to who can think of the highest number



# Single-item auctions

- General structure of an auction:
  - **Input:** every agent  $i$  submits a bid  $b_i$  (*agents = bidders*)
  - **Allocation rule:** decide the winner
  - **Payment rule:** decide a selling price
- Deciding the winner is easy: the highest bidder
- Deciding the selling price is more complicated
  - A selling price of 0, creates a competition among the bidders as to who can think of the highest number
- We are interested in payment rules that incentivize the bidders to bid their true values

# Single-item auctions

- General structure of an auction:
  - **Input:** every agent  $i$  submits a bid  $b_i$  (*agents = bidders*)
  - **Allocation rule:** decide the winner
  - **Payment rule:** decide a selling price
- Deciding the winner is easy: the highest bidder
- Deciding the selling price is more complicated
  - A selling price of 0, creates a competition among the bidders as to who can think of the highest number
- We are interested in payment rules that incentivize the bidders to bid their true values
  - **Truthful auctions that maximize the social welfare**

# First-price auction

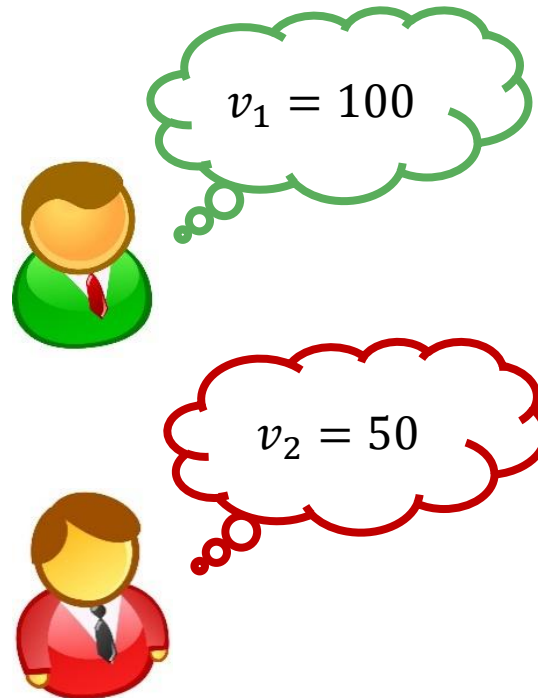
- **Allocation rule:** the winner is the highest bidder
- **Payment rule:** the winner pays her bid

# First-price auction

- **Allocation rule:** the winner is the highest bidder
- **Payment rule:** the winner pays her bid
- Is this a truthful auction?

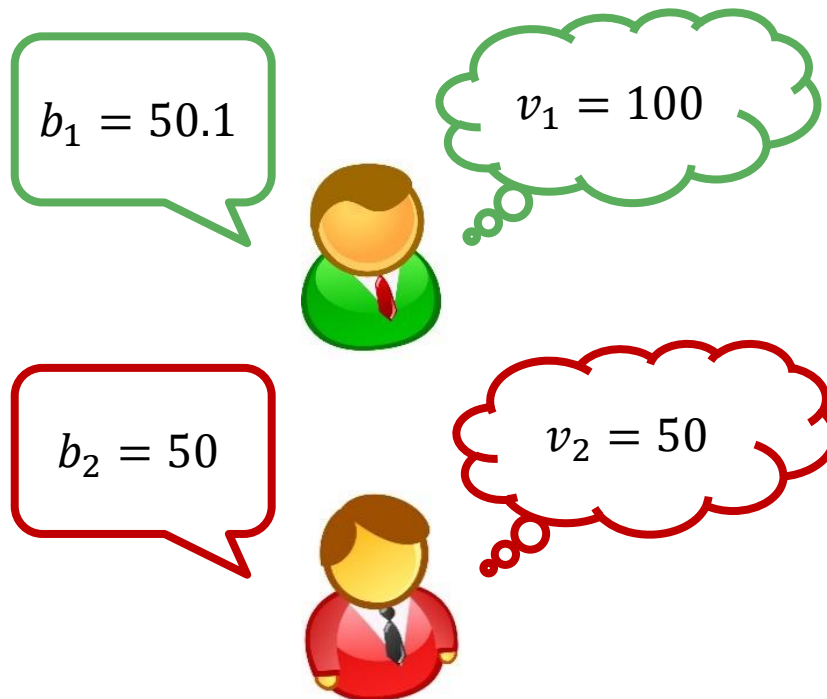
# First-price auction

- **Allocation rule:** the winner is the highest bidder
- **Payment rule:** the winner pays her bid
- Is this a truthful auction?



# First-price auction

- **Allocation rule:** the winner is the highest bidder
- **Payment rule:** the winner pays her bid
- Is this a truthful auction?



# Second-price auction

- **Allocation rule:** the winner is the highest bidder
- **Payment rule:** the winner pays the second highest bid

# Second-price auction

- **Allocation rule:** the winner is the highest bidder
- **Payment rule:** the winner pays the second highest bid

## Theorem [Vickrey, 1961]

In a second-price auction

- (a) it is a dominant strategy for every bidder  $i$  to bid  $b_i = v_i$ , and
- (b) every truthtelling bidder gets non-negative utility



# Second-price auction

- **Allocation rule:** the winner is the highest bidder
- **Payment rule:** the winner pays the second highest bid

## Theorem [Vickrey, 1961]

In a second-price auction

- (a) it is a dominant strategy for every bidder  $i$  to bid  $b_i = v_i$ , and
- (b) every truthtelling bidder gets non-negative utility

- (b) is obvious:
  - the selling price is at most the winner's bid, and the bid of a truthtelling bidder is equal to her true value

# Second-price auction

- For (a), our goal is to show that the utility of bidder  $i$  is maximized by bidding  $v_i$ , no matter what  $v_i$  and the bids of the other bidders are

# Second-price auction

- For (a), our goal is to show that the utility of bidder  $i$  is maximized by bidding  $v_i$ , no matter what  $v_i$  and the bids of the other bidders are
- Second highest bid:  $B = \max_{j \neq i} b_j$
- The utility of bidder  $i$  is either 0 if  $b_i < B$ , or  $v_i - B$  otherwise

# Second-price auction

- For (a), our goal is to show that the utility of bidder  $i$  is maximized by bidding  $v_i$ , no matter what  $v_i$  and the bids of the other bidders are
- Second highest bid:  $B = \max_{j \neq i} b_j$
- The utility of bidder  $i$  is either 0 if  $b_i < B$ , or  $v_i - B$  otherwise

## Case I: $v_i < B$

- Maximum possible utility = 0
- Achieved by setting  $b_i = v_i$

# Second-price auction

- For (a), our goal is to show that the utility of bidder  $i$  is maximized by bidding  $v_i$ , no matter what  $v_i$  and the bids of the other bidders are
- Second highest bid:  $B = \max_{j \neq i} b_j$
- The utility of bidder  $i$  is either 0 if  $b_i < B$ , or  $v_i - B$  otherwise

## Case I: $v_i < B$

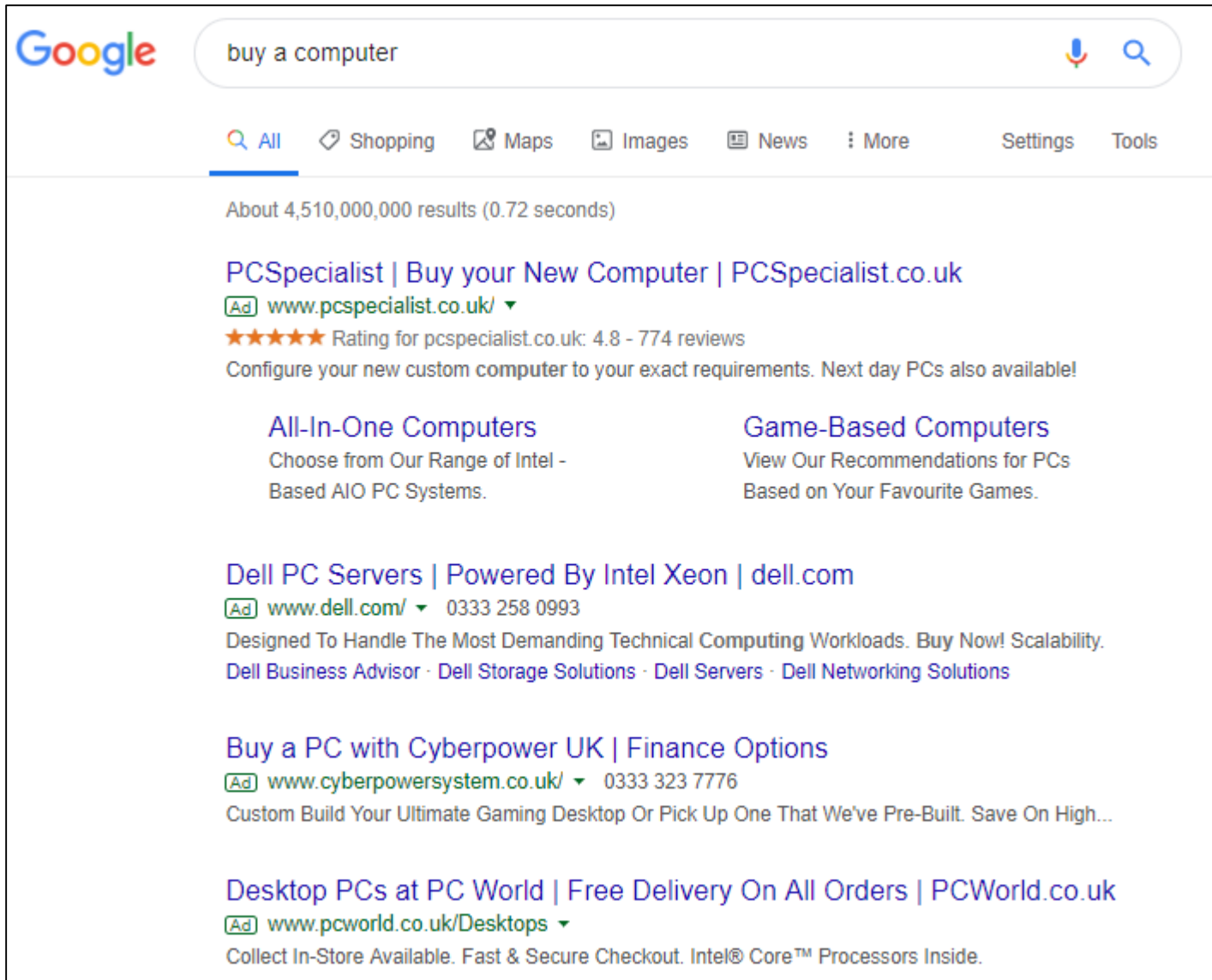
- Maximum possible utility = 0
- Achieved by setting  $b_i = v_i$

## Case II: $v_i \geq B$

- Maximum possible utility =  $v_i - B$
- Bidder  $i$  wins the item by setting  $b_i = v_i$



# Sponsored search auctions



The image shows a Google search results page for the query "buy a computer". The search bar at the top contains the text "buy a computer" and has a microphone icon and a search icon to its right. Below the search bar, there are navigation links for "All", "Shopping", "Maps", "Images", "News", "More", "Settings", and "Tools". The search results show "About 4,510,000,000 results (0.72 seconds)".

The first sponsored result is from PCSpecialist.co.uk. The ad title is "PCSpecialist | Buy your New Computer | PCSpecialist.co.uk". Below the title, there is a green "Ad" label, the URL "www.pcspecialist.co.uk/", and a star rating of 4.8 from 774 reviews. The ad text says "Configure your new custom computer to your exact requirements. Next day PCs also available!". There are two columns of text below the ad: "All-In-One Computers" with the subtext "Choose from Our Range of Intel - Based AIO PC Systems." and "Game-Based Computers" with the subtext "View Our Recommendations for PCs Based on Your Favourite Games."

The second sponsored result is from Dell.com. The ad title is "Dell PC Servers | Powered By Intel Xeon | dell.com". Below the title, there is a green "Ad" label, the URL "www.dell.com/", and the phone number "0333 258 0993". The ad text says "Designed To Handle The Most Demanding Technical Computing Workloads. Buy Now! Scalability. Dell Business Advisor · Dell Storage Solutions · Dell Servers · Dell Networking Solutions".

The third sponsored result is from Cyberpower UK. The ad title is "Buy a PC with Cyberpower UK | Finance Options". Below the title, there is a green "Ad" label, the URL "www.cyberpowersystem.co.uk/", and the phone number "0333 323 7776". The ad text says "Custom Build Your Ultimate Gaming Desktop Or Pick Up One That We've Pre-Built. Save On High...".

The fourth sponsored result is from PCWorld.co.uk. The ad title is "Desktop PCs at PC World | Free Delivery On All Orders | PCWorld.co.uk". Below the title, there is a green "Ad" label, the URL "www.pcworld.co.uk/Desktops", and the text "Collect In-Store Available. Fast & Secure Checkout. Intel® Core™ Processors Inside."

# Sponsored search auctions

- $k$  advertising **slots**
- $n$  **bidders** (advertisers) who aim to occupy a slot

# Sponsored search auctions

- $k$  advertising **slots**
- $n$  **bidders** (advertisers) who aim to occupy a slot
- Slot  $j$  has a **click-through-rate** (CTR)  $a_j$ 
  - The CTR of a slot represents the **probability that the ad placed at this slot will be clicked on**
  - Assumption: the CTRs are independent of the ads that occupy the slots



# Sponsored search auctions

- $k$  advertising **slots**
- $n$  **bidders** (advertisers) who aim to occupy a slot
- Slot  $j$  has a **click-through-rate** (CTR)  $a_j$ 
  - The CTR of a slot represents the **probability that the ad placed at this slot will be clicked on**
  - Assumption: the CTRs are independent of the ads that occupy the slots
- The slots are ranked so that  $a_1 \geq \dots \geq a_k$

# Sponsored search auctions

- $k$  advertising **slots**
- $n$  **bidders** (advertisers) who aim to occupy a slot
- Slot  $j$  has a **click-through-rate** (CTR)  $a_j$ 
  - The CTR of a slot represents the **probability that the ad placed at this slot will be clicked on**
  - Assumption: the CTRs are independent of the ads that occupy the slots
- The slots are ranked so that  $a_1 \geq \dots \geq a_k$
- Each bidder  $i$  has a **private value**  $v_i$  **per click**
  - Bidder  $i$  derives utility  $a_j \cdot v_i$  from slot  $j$

# Sponsored search auctions: goals

- **Truthfulness:** It is a dominant strategy for each bidder to bid her true value
- **Social welfare maximization:**  $\sum_i v_i \cdot x_i$ 
  - $x_i$  is the CTR of the slot that bidder  $i$  is assigned to, or 0 otherwise
- **Poly-time execution:** running the auction should be quick

# Sponsored search auctions: goals

- **Truthfulness:** It is a dominant strategy for each bidder to bid her true value
- **Social welfare maximization:**  $\sum_i v_i \cdot x_i$ 
  - $x_i$  is the CTR of the slot that bidder  $i$  is assigned to, or 0 otherwise
- **Poly-time execution:** running the auction should be quick
- If the bidders are truthful, then maximizing the social welfare is easy: sort the bidders in decreasing order of their bids
- So, the problem is to incentivize them to be truthful, again

# Sponsored search auctions: goals

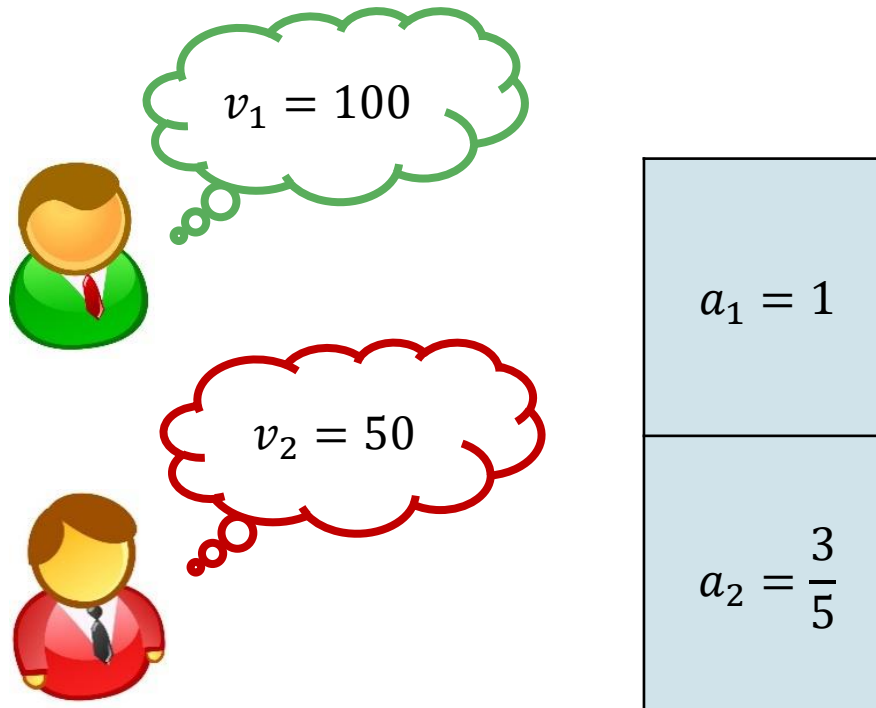
- **Truthfulness:** It is a dominant strategy for each bidder to bid her true value
- **Social welfare maximization:**  $\sum_i v_i \cdot x_i$ 
  - $x_i$  is the CTR of the slot that bidder  $i$  is assigned to, or 0 otherwise
- **Poly-time execution:** running the auction should be quick
- If the bidders are truthful, then maximizing the social welfare is easy: sort the bidders in decreasing order of their bids
- So, the problem is to incentivize them to be truthful, again
- Can we extend the ideas we exploited for single-item auctions?

# Generalized second-price auction

- **Allocation rule:** sort the bidders in decreasing order of their bids and rename them so that  $b_1 \geq \dots \geq b_n$
- **Payment rule:** every bidder  $i \leq k$  (who is assigned at slot  $i$ ) pays the next highest bid  $b_{i+1}$  per click, and every bidder  $i > k$  pays 0

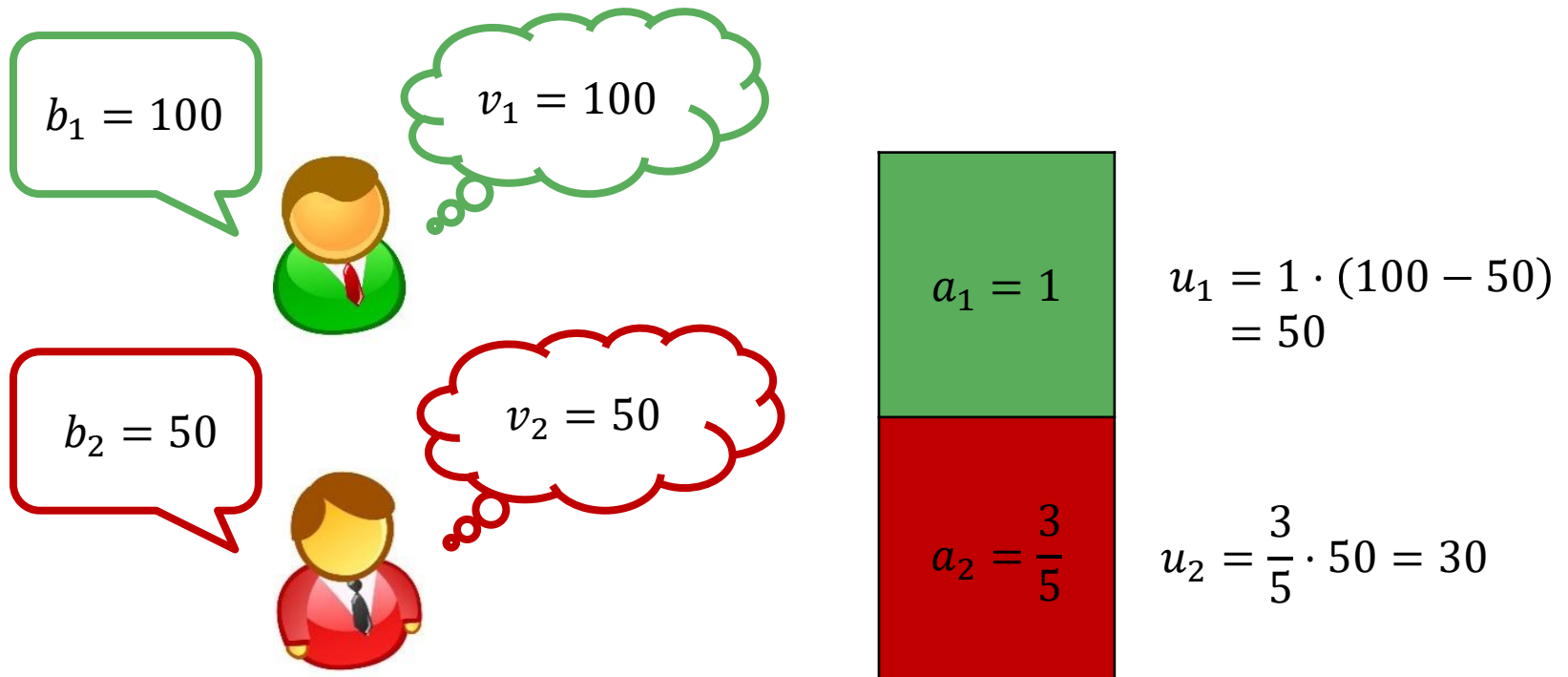
# Generalized second-price auction

- **Allocation rule:** sort the bidders in decreasing order of their bids and rename them so that  $b_1 \geq \dots \geq b_n$
- **Payment rule:** every bidder  $i \leq k$  (who is assigned at slot  $i$ ) pays the next highest bid  $b_{i+1}$  per click, and every bidder  $i > k$  pays 0



# Generalized second-price auction

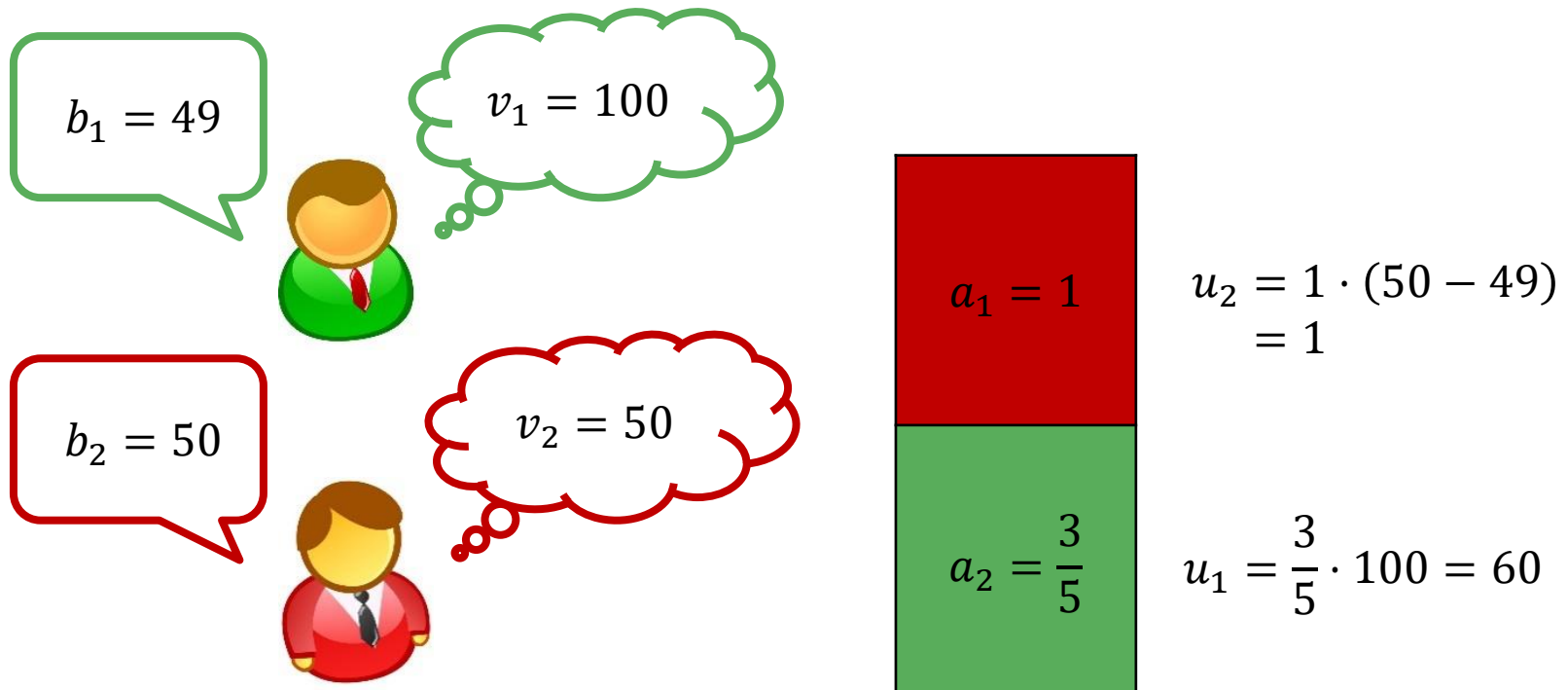
- **Allocation rule:** sort the bidders in decreasing order of their bids and rename them so that  $b_1 \geq \dots \geq b_n$
- **Payment rule:** every bidder  $i \leq k$  (who is assigned at slot  $i$ ) pays the next highest bid  $b_{i+1}$  per click, and every bidder  $i > k$  pays 0





# Generalized second-price auction

- **Allocation rule:** sort the bidders in decreasing order of their bids and rename them so that  $b_1 \geq \dots \geq b_n$
- **Payment rule:** every bidder  $i \leq k$  (who is assigned at slot  $i$ ) pays the next highest bid  $b_{i+1}$  per click, and every bidder  $i > k$  pays 0



# Myerson's Lemma

- That didn't work for sponsored search auctions, so what now?

# Myerson's Lemma

- That didn't work for sponsored search auctions, so what now?
- Let's try to see how the optimal truthful auction should look like, for any single parameter environment

# Myerson's Lemma

- That didn't work for sponsored search auctions, so what now?
- Let's try to see how the optimal truthful auction should look like, for any single parameter environment
- Input by bidders:  $\mathbf{b} = (b_1, \dots, b_n)$
- Allocation rule:  $\mathbf{x}(\mathbf{b}) = (x_1(\mathbf{b}), \dots, x_n(\mathbf{b}))$
- Payment rule:  $\mathbf{p}(\mathbf{b}) = (p_1(\mathbf{b}), \dots, p_n(\mathbf{b}))$
- The utility of bidder  $i$  is  $u_i(\mathbf{b}) = v_i \cdot x_i(\mathbf{b}) - p_i(\mathbf{b})$

# Myerson's Lemma

- That didn't work for sponsored search auctions, so what now?
- Let's try to see how the optimal truthful auction should look like, for any single parameter environment
- Input by bidders:  $\mathbf{b} = (b_1, \dots, b_n)$
- Allocation rule:  $\mathbf{x}(\mathbf{b}) = (x_1(\mathbf{b}), \dots, x_n(\mathbf{b}))$
- Payment rule:  $\mathbf{p}(\mathbf{b}) = (p_1(\mathbf{b}), \dots, p_n(\mathbf{b}))$
- The utility of bidder  $i$  is  $u_i(\mathbf{b}) = v_i \cdot x_i(\mathbf{b}) - p_i(\mathbf{b})$
- Focus on payment rules such that  $p_i(\mathbf{b}) \in [0, b_i \cdot x_i(\mathbf{b})]$ 
  - $p_i(\mathbf{b}) \geq 0$  ensures that the seller does not pay the bidders
  - $p_i(\mathbf{b}) \leq b_i \cdot x_i(\mathbf{b})$  ensures non-negative utility for truthful bidders

# Myerson's Lemma

- An allocation rule  $x$  is **implementable** if there exists a payment rule  $p$  such that  $(x, p)$  is a truthful auction

# Myerson's Lemma

- An allocation rule  $x$  is **implementable** if there exists a payment rule  $p$  such that  $(x, p)$  is a truthful auction
- An allocation rule  $x$  is **monotone** if for every bidder  $i$  and bid vector  $\mathbf{b}_{-i}$ , the allocation  $x_i(z, \mathbf{b}_{-i})$  is non-decreasing in the bid  $z$  of bidder  $i$

# Myerson's Lemma

- An allocation rule  $x$  is **implementable** if there exists a payment rule  $p$  such that  $(x, p)$  is a truthful auction
- An allocation rule  $x$  is **monotone** if for every bidder  $i$  and bid vector  $\mathbf{b}_{-i}$ , the allocation  $x_i(z, \mathbf{b}_{-i})$  is non-decreasing in the bid  $z$  of bidder  $i$

## Lemma [Myerson, 1981]

- (a) An allocation rule  $x$  is implementable if and only if it is monotone
- (b) For every allocation rule  $x$ , there exists a unique payment rule  $p$  such that  $(x, p)$  is a truthful auction



# Proof of Myerson's Lemma

- Fix a bidder  $i$ , and the bids  $\mathbf{b}_{-i}$  of the other bidders
- Given that these quantities are now fixed, we simplify our notation:
  - $x(z) = x_i(z, \mathbf{b}_{-i})$
  - $p(z) = p_i(z, \mathbf{b}_{-i})$
  - $u(z) = u_i(z, \mathbf{b}_{-i})$

# Proof of Myerson's Lemma

- Fix a bidder  $i$ , and the bids  $\mathbf{b}_{-i}$  of the other bidders
- Given that these quantities are now fixed, we simplify our notation:
  - $x(z) = x_i(z, \mathbf{b}_{-i})$
  - $p(z) = p_i(z, \mathbf{b}_{-i})$
  - $u(z) = u_i(z, \mathbf{b}_{-i})$
- The idea:
  - assuming  $(\mathbf{x}, \mathbf{p})$  is a truthful auction, the bidder has no incentive to unilaterally deviate to any other bid
  - This will give us a relation between  $\mathbf{x}$  and  $\mathbf{p}$ , which we can use to derive an explicit formula for  $\mathbf{p}$  as a function of  $\mathbf{x}$

# Proof of Myerson's Lemma

- Consider two bids  $0 \leq z < y$  and assume  $x$  is implementable by  $p$

# Proof of Myerson's Lemma

- Consider two bids  $0 \leq z < y$  and assume  $x$  is implementable by  $p$
- True value =  $z$ , deviating bid =  $y$ :

$$u(z) \geq u(y)$$

# Proof of Myerson's Lemma

- Consider two bids  $0 \leq z < y$  and assume  $x$  is implementable by  $p$
- True value =  $z$ , deviating bid =  $y$ :

$$u(z) \geq u(y) \Leftrightarrow z \cdot x(z) - p(z) \geq z \cdot x(y) - p(y)$$

# Proof of Myerson's Lemma

- Consider two bids  $0 \leq z < y$  and assume  $x$  is implementable by  $p$
- True value =  $z$ , deviating bid =  $y$ :

$$\begin{aligned}u(z) \geq u(y) &\Leftrightarrow z \cdot x(z) - p(z) \geq z \cdot x(y) - p(y) \\ &\Leftrightarrow p(y) - p(z) \geq z \cdot (x(y) - x(z))\end{aligned}$$

# Proof of Myerson's Lemma

- Consider two bids  $0 \leq z < y$  and assume  $x$  is implementable by  $p$
- True value =  $z$ , deviating bid =  $y$ :

$$\begin{aligned}u(z) \geq u(y) &\Leftrightarrow z \cdot x(z) - p(z) \geq z \cdot x(y) - p(y) \\ &\Leftrightarrow p(y) - p(z) \geq z \cdot (x(y) - x(z))\end{aligned}$$

- True value =  $y$ , deviating bid =  $z$ :

$$\begin{aligned}u(y) \geq u(z) &\Leftrightarrow y \cdot x(y) - p(y) \geq y \cdot x(z) - p(z) \\ &\Leftrightarrow p(y) - p(z) \leq y \cdot (x(y) - x(z))\end{aligned}$$

# Proof of Myerson's Lemma

- Combining these two, we get:

$$z \cdot (x(y) - x(z)) \leq p(y) - p(z) \leq y \cdot (x(y) - x(z))$$



# Proof of Myerson's Lemma

- Combining these two, we get:

$$z \cdot (x(y) - x(z)) \leq p(y) - p(z) \leq y \cdot (x(y) - x(z))$$

- This also implies that

$$(y - z) \cdot (x(y) - x(z)) \geq 0$$

# Proof of Myerson's Lemma

- Combining these two, we get:

$$z \cdot (x(y) - x(z)) \leq p(y) - p(z) \leq y \cdot (x(y) - x(z))$$

- This also implies that

$$(y - z) \cdot (x(y) - x(z)) \geq 0$$

- Since  $0 \leq z < y$ , this is possible if and only if  $x$  is monotone so that  $y - z \leq 0$  and  $x(y) - x(z) \leq 0$

# Proof of Myerson's Lemma

- Combining these two, we get:

$$z \cdot (x(y) - x(z)) \leq p(y) - p(z) \leq y \cdot (x(y) - x(z))$$

- This also implies that

$$(y - z) \cdot (x(y) - x(z)) \geq 0$$

- Since  $0 \leq z < y$ , this is possible if and only if  $x$  is monotone so that  $y - z \leq 0$  and  $x(y) - x(z) \leq 0$

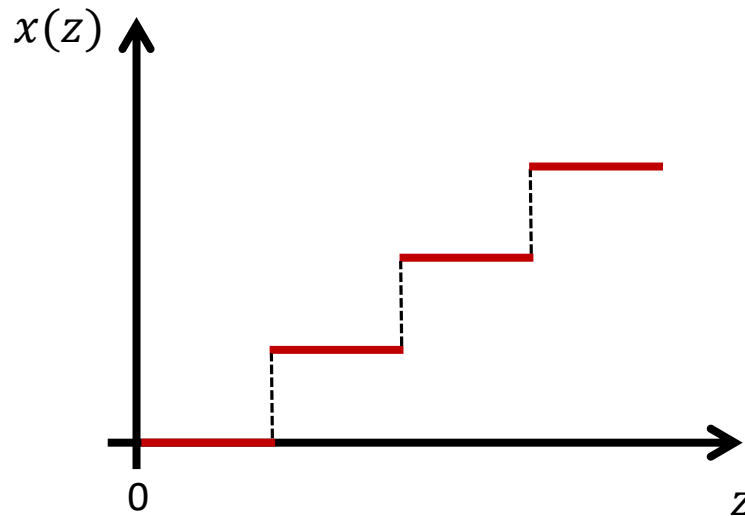
$\Rightarrow$  (a) is now proved

# Proof of Myerson's Lemma

- We can now assume that  $x$  is monotone

# Proof of Myerson's Lemma

- We can now assume that  $x$  is monotone
- Assume  $x$  is piecewise constant, like in sponsored search auctions



- The break points are defined by the highest bids of the other bidders

# Proof of Myerson's Lemma

$$z \cdot (x(y) - x(z)) \leq p(y) - p(z) \leq y \cdot (x(y) - x(z))$$

# Proof of Myerson's Lemma

$$z \cdot (x(y) - x(z)) \leq p(y) - p(z) \leq y \cdot (x(y) - x(z))$$

- By fixing  $z$  and taking the limit as  $y$  tends to  $z$ , we have that

$$\text{jump of } p \text{ at } z = z \cdot (\text{jump of } x \text{ at } z)$$

# Proof of Myerson's Lemma

$$z \cdot (x(y) - x(z)) \leq p(y) - p(z) \leq y \cdot (x(y) - x(z))$$

- By fixing  $z$  and taking the limit as  $y$  tends to  $z$ , we have that

$$\text{jump of } p \text{ at } z = z \cdot (\text{jump of } x \text{ at } z)$$

- Therefore, we can define the payment of the bidder as

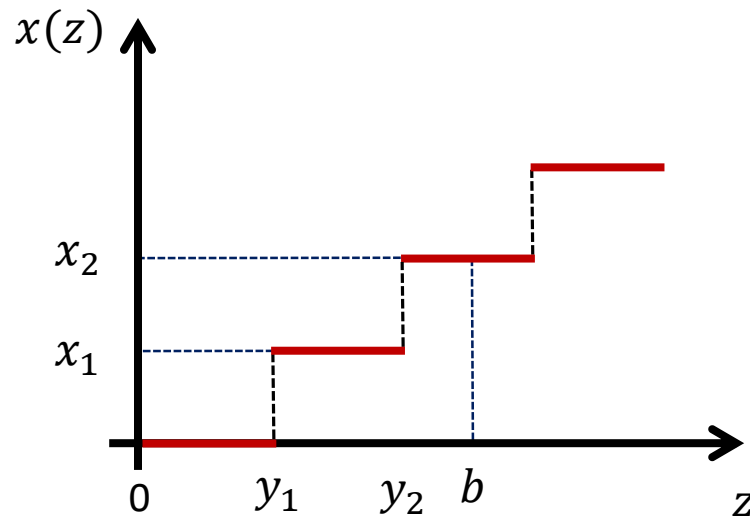
$$p(b) = \sum_{y \in [0, b]} y \cdot (\text{jump of } x \text{ at } y)$$

where  $y$  enumerates all break points of  $x$  in  $[0, b]$



# Proof of Myerson's Lemma

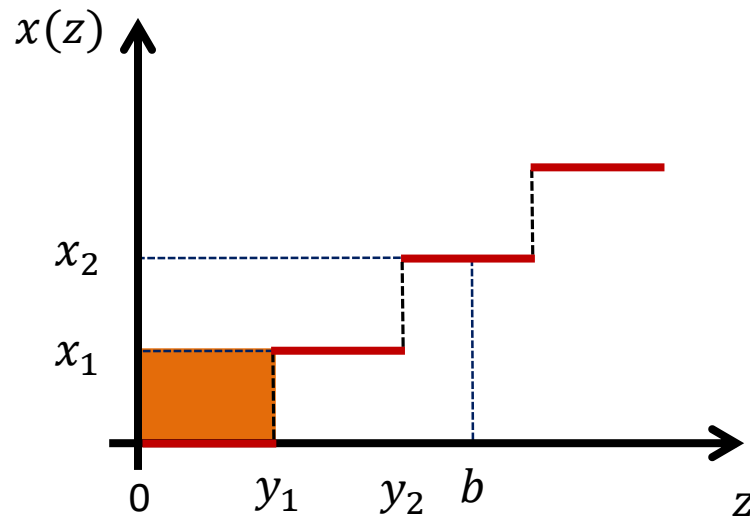
- Example:



$$p(b) = \sum_{y \in [0, b]} y \cdot (\text{jump of } x \text{ at } y)$$

# Proof of Myerson's Lemma

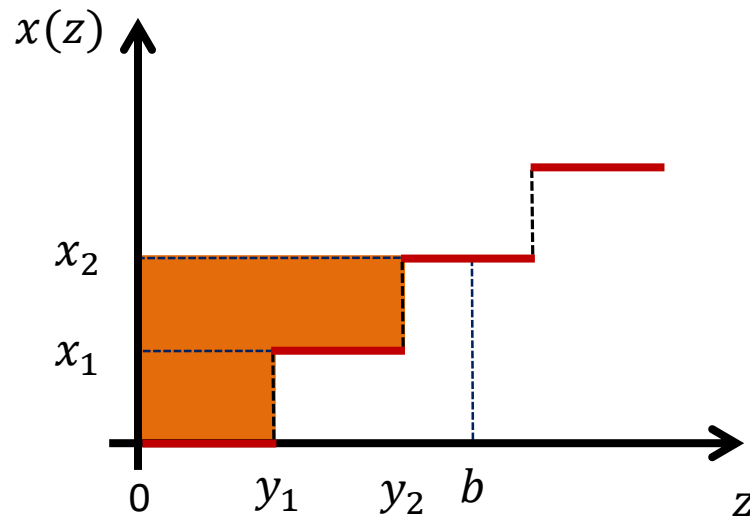
- Example:



$$p(b) = \sum_{y \in [0, b]} y \cdot (\text{jump of } x \text{ at } y) = y_1 \cdot x_1$$

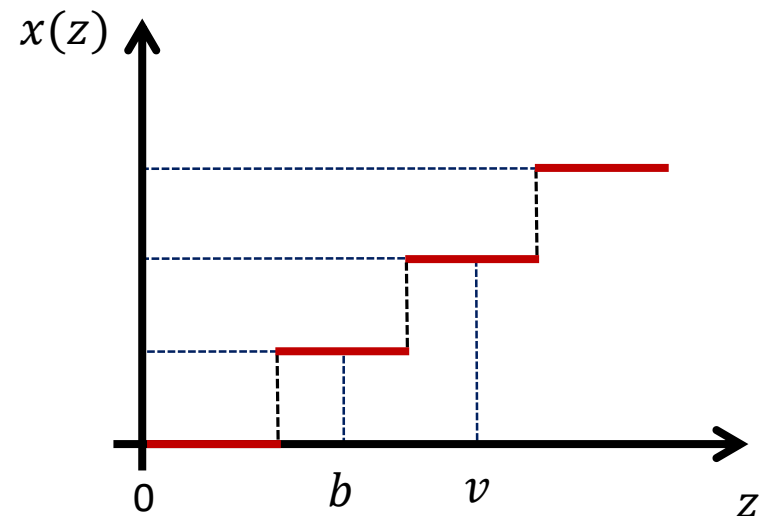
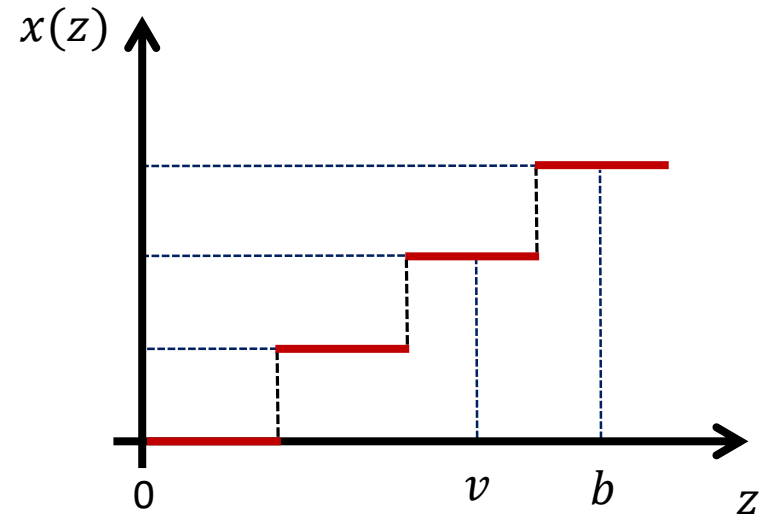
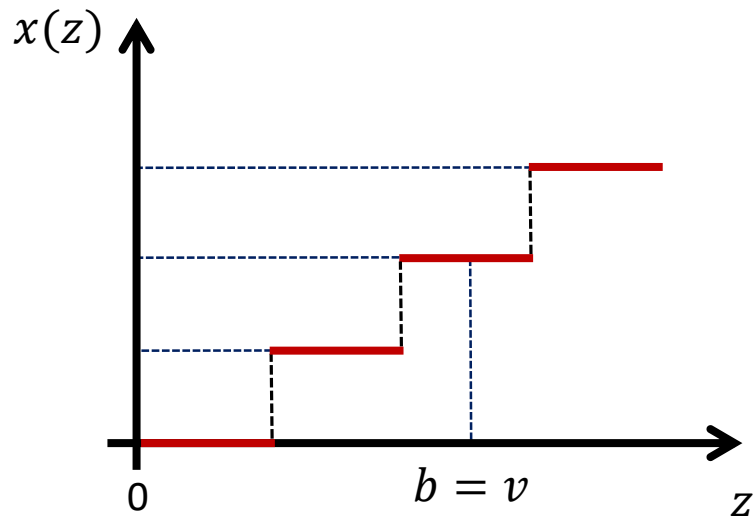
# Proof of Myerson's Lemma

- Example:

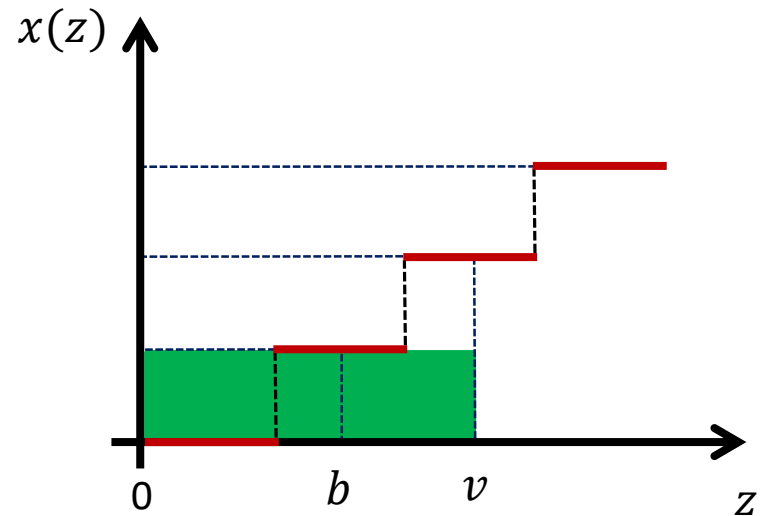
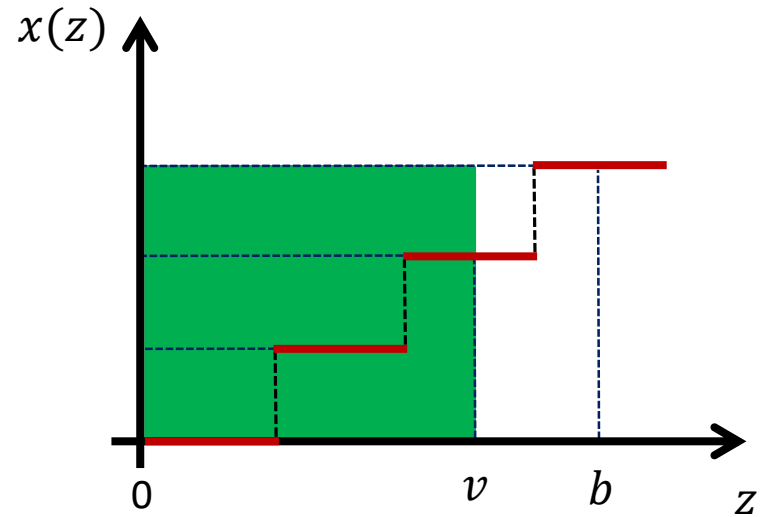
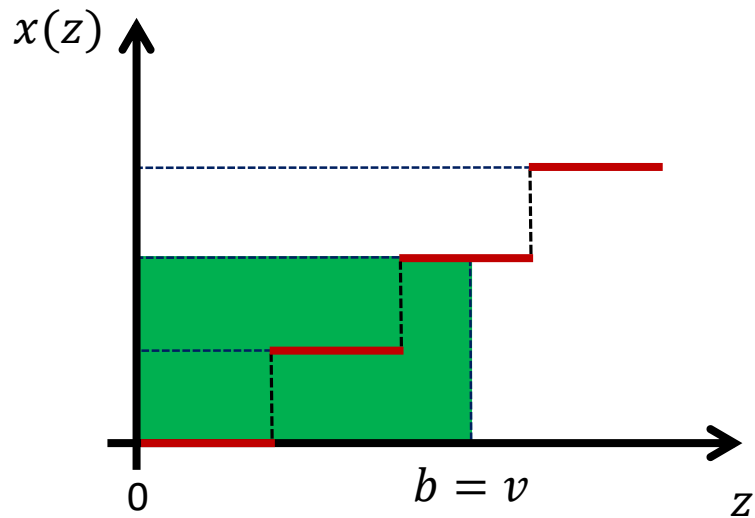


$$p(b) = \sum_{y \in [0, b]} y \cdot (\text{jump of } x \text{ at } y) = y_1 \cdot x_1 + y_2 \cdot (x_2 - x_1)$$

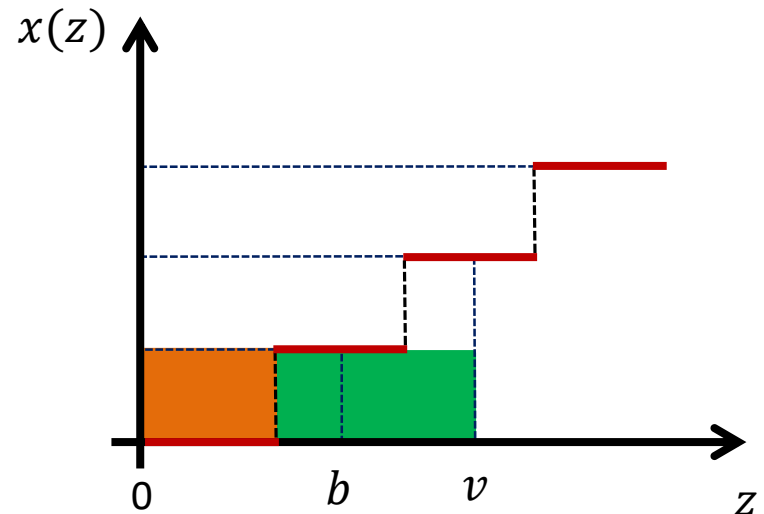
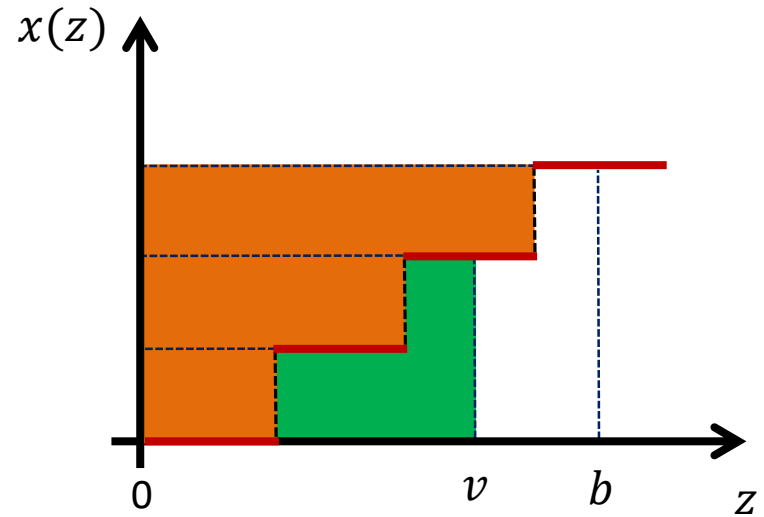
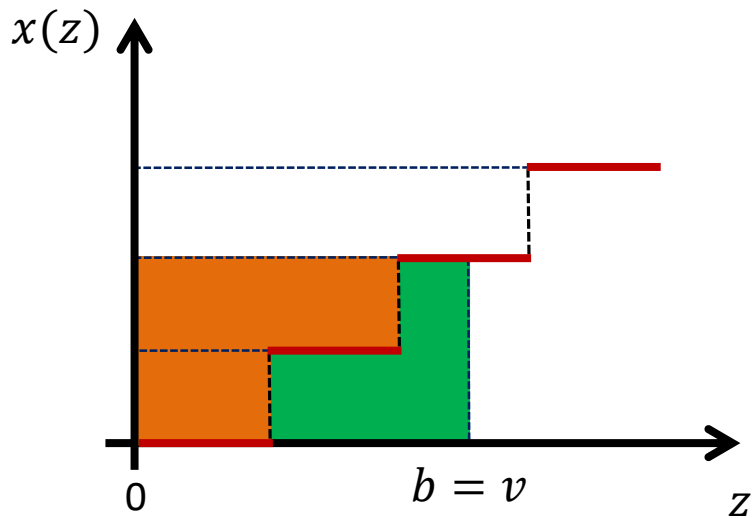
# Proof of Myerson's Lemma



# Proof of Myerson's Lemma



# Proof of Myerson's Lemma



# Sponsored search auctions

$$p(b) = \sum_{y \in [0, b]} y \cdot (\text{jump of } x \text{ at } y)$$

# Sponsored search auctions

$$p(b) = \sum_{y \in [0, b]} y \cdot (\text{jump of } x \text{ at } y)$$

- $y$  enumerates the break points: the bids that are smaller than  $b$ 
  - In other words,  $y$  enumerates the slots from worst to best



# Sponsored search auctions

$$p(b) = \sum_{y \in [0, b]} y \cdot (\text{jump of } x \text{ at } y)$$

- $y$  enumerates the break points: the bids that are smaller than  $b$ 
  - In other words,  $y$  enumerates the slots from worst to best
- jump of  $x$  at  $y$ : the difference in CTR between two consecutive slots

# Sponsored search auctions

$$p(b) = \sum_{y \in [0, b]} y \cdot (\text{jump of } x \text{ at } y)$$

- $y$  enumerates the break points: the bids that are smaller than  $b$ 
  - In other words,  $y$  enumerates the slots from worst to best
- jump of  $x$  at  $y$ : the difference in CTR between two consecutive slots
- The total payment of the  $i$ -th highest bidder is:

$$p_i(b_i, \mathbf{b}_{-i}) = \sum_{j=i}^k b_{j+1} (a_j - a_{j+1})$$

# Summary

# Summary

- **Auctions:** allocation rule + payment rule

# Summary

- **Auctions:** allocation rule + payment rule
- An allocation rule is **implementable** if there exists a payment rule, so that together they define a truthful auction

# Summary

- **Auctions:** allocation rule + payment rule
- An allocation rule is **implementable** if there exists a payment rule, so that together they define a truthful auction
- An allocation rule is **monotone**, if larger bids give more stuff

# Summary

- **Auctions:** allocation rule + payment rule
- An allocation rule is **implementable** if there exists a payment rule, so that together they define a truthful auction
- An allocation rule is **monotone**, if larger bids give more stuff
- **Single-item auctions:** first-price is not truthful, second-price is truthful and maximizes the social welfare (sells to the bidder with the highest value)

# Summary

- **Auctions:** allocation rule + payment rule
- An allocation rule is **implementable** if there exists a payment rule, so that together they define a truthful auction
- An allocation rule is **monotone**, if larger bids give more stuff
- **Single-item auctions:** first-price is not truthful, second-price is truthful and maximizes the social welfare (sells to the bidder with the highest value)
- **Sponsored search auctions:** generalized second-price auction is not truthful



# Summary

- **Auctions:** allocation rule + payment rule
- An allocation rule is **implementable** if there exists a payment rule, so that together they define a truthful auction
- An allocation rule is **monotone**, if larger bids give more stuff
- **Single-item auctions:** first-price is not truthful, second-price is truthful and maximizes the social welfare (sells to the bidder with the highest value)
- **Sponsored search auctions:** generalized second-price auction is not truthful
- **Myerson's Lemma:** a characterization of truthful mechanisms in single-parameter environments

# Summary

- **Auctions:** allocation rule + payment rule
- An allocation rule is **implementable** if there exists a payment rule, so that together they define a truthful auction
- An allocation rule is **monotone**, if larger bids give more stuff
- **Single-item auctions:** first-price is not truthful, second-price is truthful and maximizes the social welfare (sells to the bidder with the highest value)
- **Sponsored search auctions:** generalized second-price auction is not truthful
- **Myerson's Lemma:** a characterization of truthful mechanisms in single-parameter environments
- Using Myerson's Lemma we can design a truthful sponsored search auction