Myerson's Lemma

- For a single parameter environment
  - An allocation rule x is implementable if and only if it is monotone
  - If x is monotone, then there exists a unique payment rule such that the sealed-bid mechanism (x, p) is DISC [assuming that $b_i = 0 \Rightarrow p_i(b) = 0$ (normalization)]
  - The payment rule is given by an explicit formula

- There is no ambiguity in how to assign payments to achieve the DISC property!

Understanding Myerson's Lemma

- We only need to have a $\boxed{\text{monotone}}$ allocation rule to come up with a DSIC mechanism

- Suppose we choose a welfare maximizing allocation, is this rule monotone?

→ Yes!

Can we always design an awesome auction in single-parameter environment?

What we want:

- DSIC
- Welfare maximization
- ~~Fast running time~~ ?

Knapsack Auctions

Not always

- n bidders, each bidder i has
  - a publicly known size w_i (duration of TV ad)
  - Private valuation (e.g., willingness to pay for its ad being shown during the Super Bowl) $w_1=5,\ w_2=2,\ w_3=3,\ w_4=7,\ w_5=3.$

$W=10.\ (1,1,1,0,0)$

- Seller has a capacity W (the length of commercial break)
- $X=?$ $(0,0,1,1,0).$ $(0,0,0,1,1)$
- Other situations : (When there are shared resources with limited capacity, we have a knapsack problem)
  - Bidders want file stored on a shared server
  - Data streams sent through a shared communication channel
  - Processes to be executed on a shared supercomputer
  - …

$\max \sum b_i x_i$

s.t. $\sum x_i w_i \le W$

$x_i \in \{0,1\}$

NP-hard.

Knapsack Auction

- k-Item auction

- Let's try to design awesome auction through two step design process
  - Step 1: Assume bidders are truthful, find the social welfare maximization allocation x
  - Step 2: Find payment p such that (x, p) is DSIC

We just want monotone x.

Greedy Algorithm (not optimal).

- Step 0: Remove bidders who have $w_i > W$
- Step 1: Sort and re-index bidders so that

$W = 10$

$\frac{7}{6} > \frac{4}{5} = \frac{4}{5}$

$$\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \cdots \geq \frac{b_n}{w_n}$$

- Step 2: Pick winners in this order until one doesn't fit, and then stop
- Return either $\underbrace{\text{Step-2 solution}}_{b_1 + \cdots + b_k}$ or $\underbrace{\text{the highest bidder}}_{\max_i b_i}$, which creates more welfare

- Polynomial time? $O(n \log n)$.
- Approximation welfare guarantee? $\frac{1}{2}$-approximation
- Monotone?

$SW(ALG) \geqslant \frac{1}{2} SW(OPT)$

Proof: $SW(ALG) = \max\{\max_i b_i, b_1 + \cdots + b_k\}$.

$SW(OPT) \leqslant b_1 + b_2 + \cdots + b_{k+1}$ (proved in HW)

$\Rightarrow SW(OPT) \leqslant 2\, SW(ALG)$.