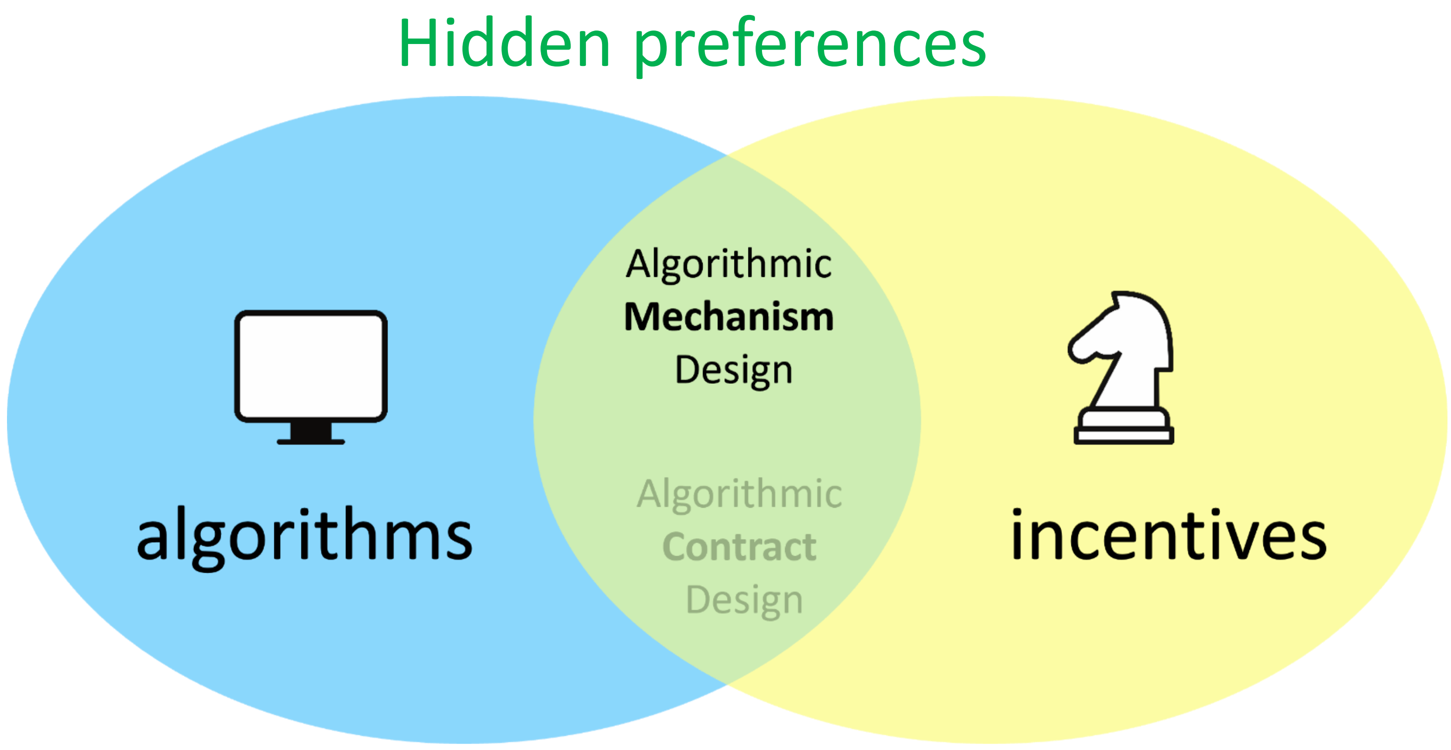
Combinatorial Contracts

Michal Feldman
Tel-Aviv University

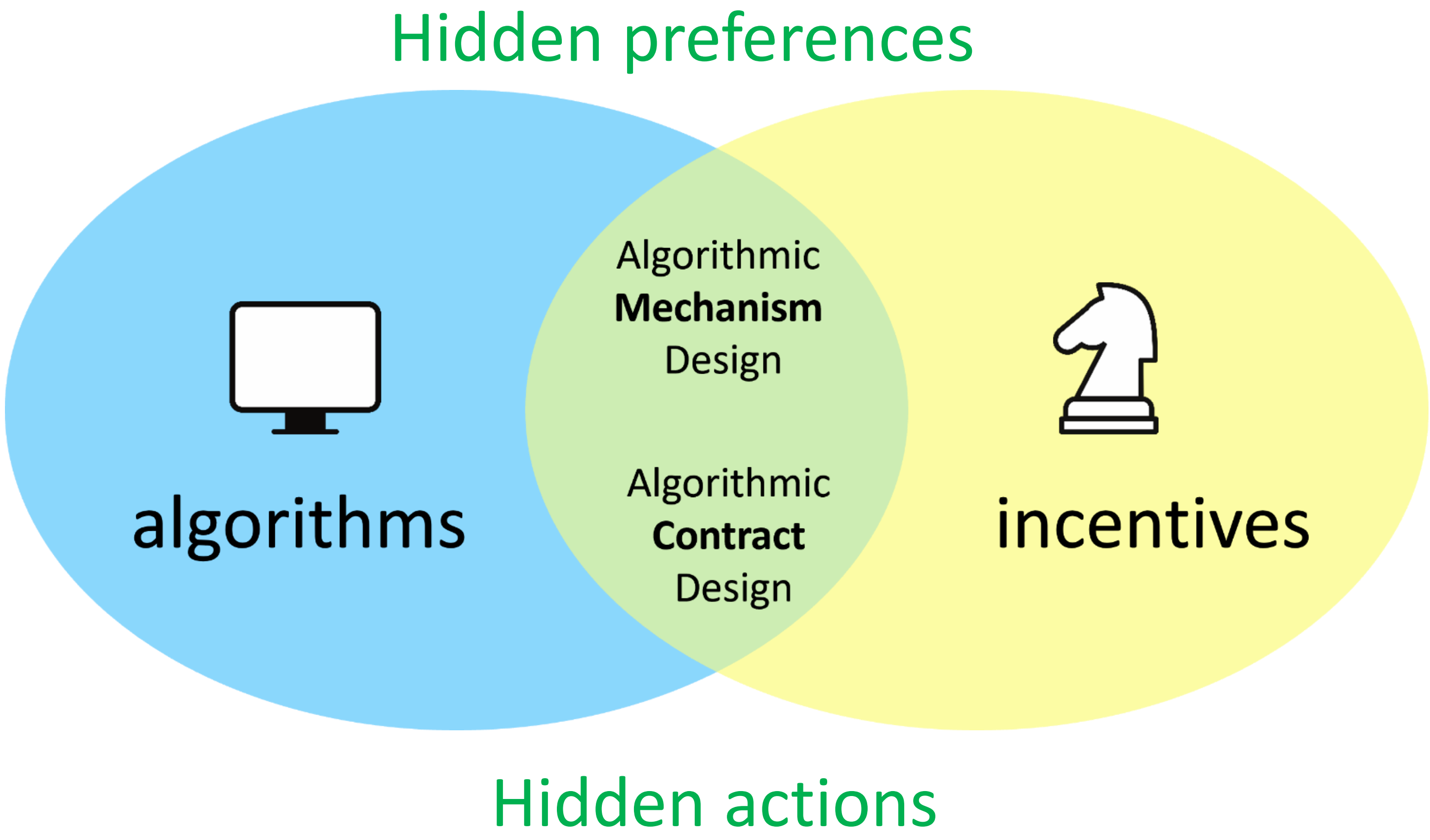
Algorithmic game theory, mechanism design, and learning
November 2022

Based on Joint Work with:
Tomer Ezra, Paul Duetting, Thomas Kesselheim (FOCS'21, Working paper)

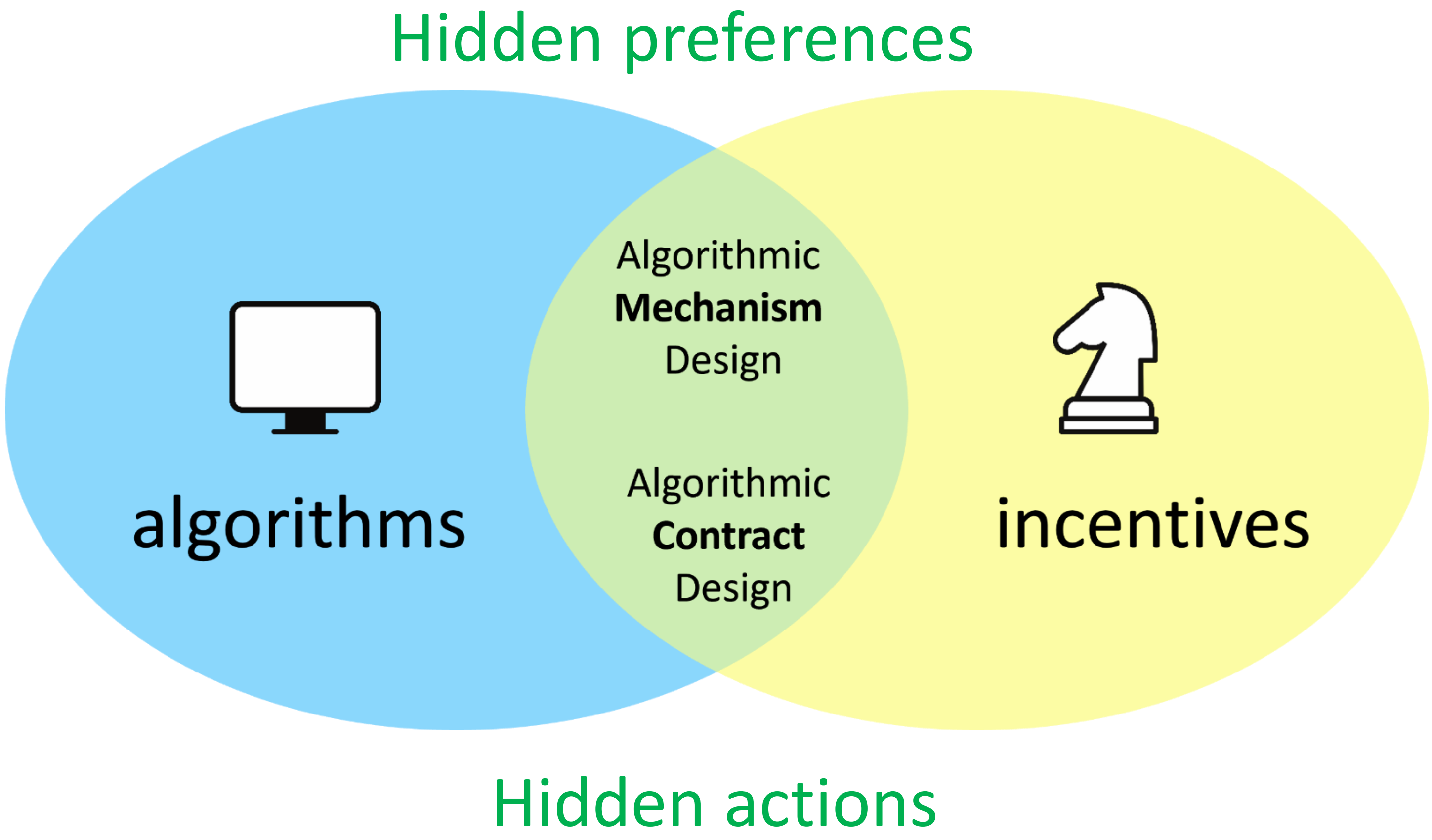
Algorithms and Incentives

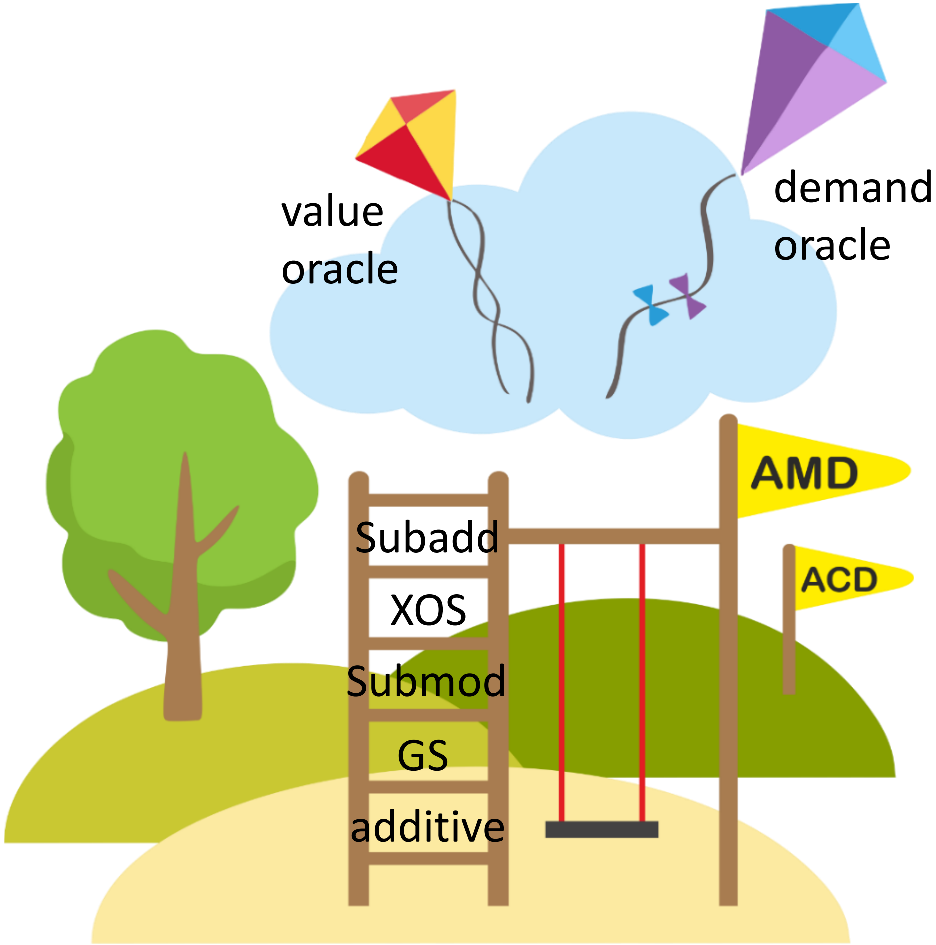


Algorithms and Incentives



Algorithms and Incentives





[Nisan Ronen 99]
[Lehmann Lehmann Nisan 01]
...

Within a Broader Perspective

An emerging frontier in Algorithmic Game Theory on optimizing the effort of others (two recent workshops in STOC'22 and EC'22)

Contracts with multiple agents / multiple actions:

[Feldman Chuang Stoica Shenker EC'05] [Babaioff Feldman Nisan EC'06] [Emek Feldman WINE'09] [Babaioff Feldman Nisan Winter JET'12] [Dütting Ezra Feldman Kesselheim FOCS'21]

Contracts with multiple outcomes:

[Dütting Roughgarden Talgam Cohen EC'19] [Dütting Roughgarden & Talgam Cohen SODA'20] [Alon Dobson Procaccia Talgam Cohen Tucker-Foltz AAAI'20] [Alon Lavi Shamash Talgam Cohen EC'21] [Alon Dütting Talgam Cohen EC'21]

Optimal scoring rules: [Chen and Yu '21] [Li et al., '22]

Delegation:

[Azar Micali TE'18] [Kleinberg Kleinberg EC'18] [Bechtel & Dughmi ITCS'21] [Braun Hahn Hoefer & Schecker '22]

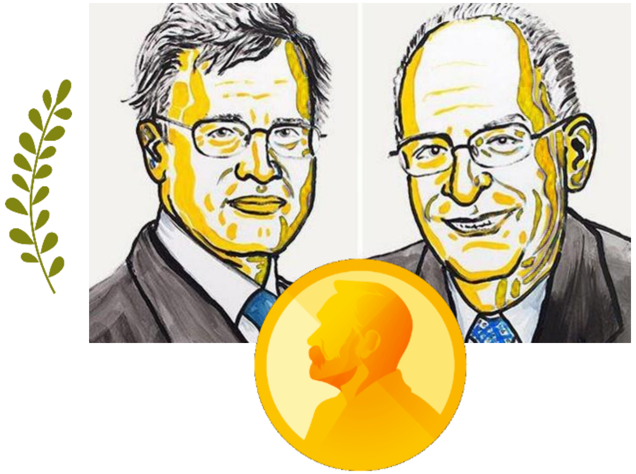
Strategic classification:

[Kleinberg & Raghavan EC'19] [Ghalme Nair Eilat Talgam Cohen Rosenfeld ICML'21] [Nair Ghalme Talgam Cohen Rosenfeld '22]

Contract Design

One of the pillars of microeconomic theory
[Ross’73, Holmstrom’79]

Holmstrom Hart



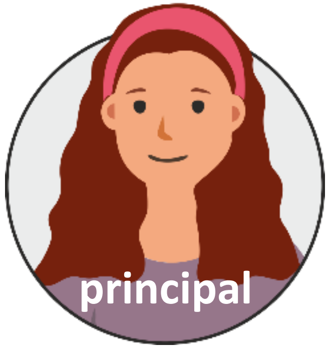
“The 2016 Nobel Prize in Economics was awarded Monday to Oliver Hart and Bengt Holmström for their work in contract theory — **developing a framework to understand agreements like insurance contracts, employer-employee relationships and property rights**.”

- As markets for services move online, they grow in scale and complexity
- An algorithmic / computational approach is timely and relevant

Principal-Agent Model

I won't be able to monitor his work. Who knows? he might go to the beach instead of focusing on the event

Organizing this event is gonna be so much work. I'll need to organize activities, do logistics, buy food, drinks, …

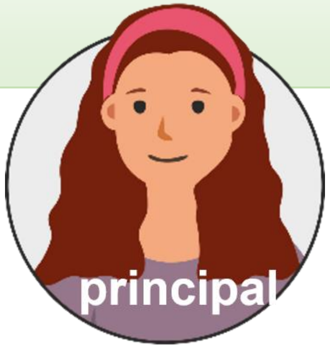


Would you please organize this event for me?

I'll only pay you if the event turns out to be a huge success

How much would you pay me?

The Principal-Agent Problem



Defines contract $\alpha \in [0,1]$

Chooses action a

Gets reward 1 with probability $f(a)$

Incurs cost $c(a)$

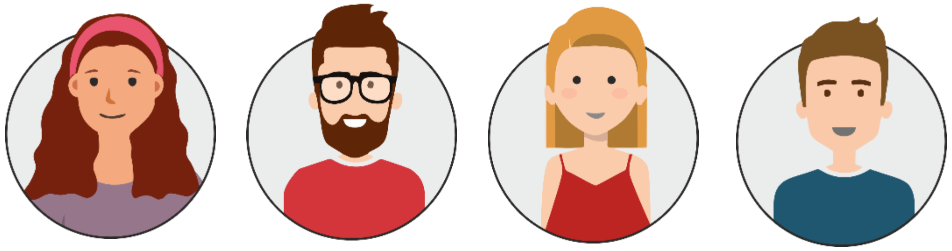
Pays $\alpha\, f(a)$

Receives $\alpha\, f(a)$

Expected utility: $(1-\alpha)f(a)$ $\alpha f(a) - c(a)$

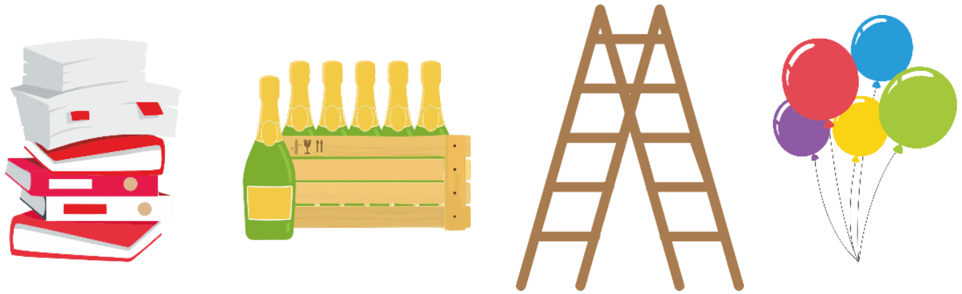
Hidden action, Stochastic outcome

Sources of Complexity in Contract Design



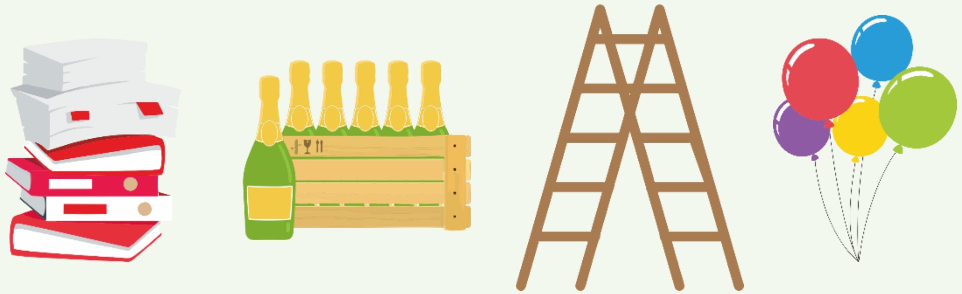
Multiple agents

[Feldman, Chuang, Stoica, Shenker EC’05, Babaioff Feldman Nisan EC’06, Emek Feldman ‘09, Ezra Duetting Feldman Kesselheim, working paper]



Multiple actions

[Ezra Duetting Feldman Kesselheim FOCS’21]



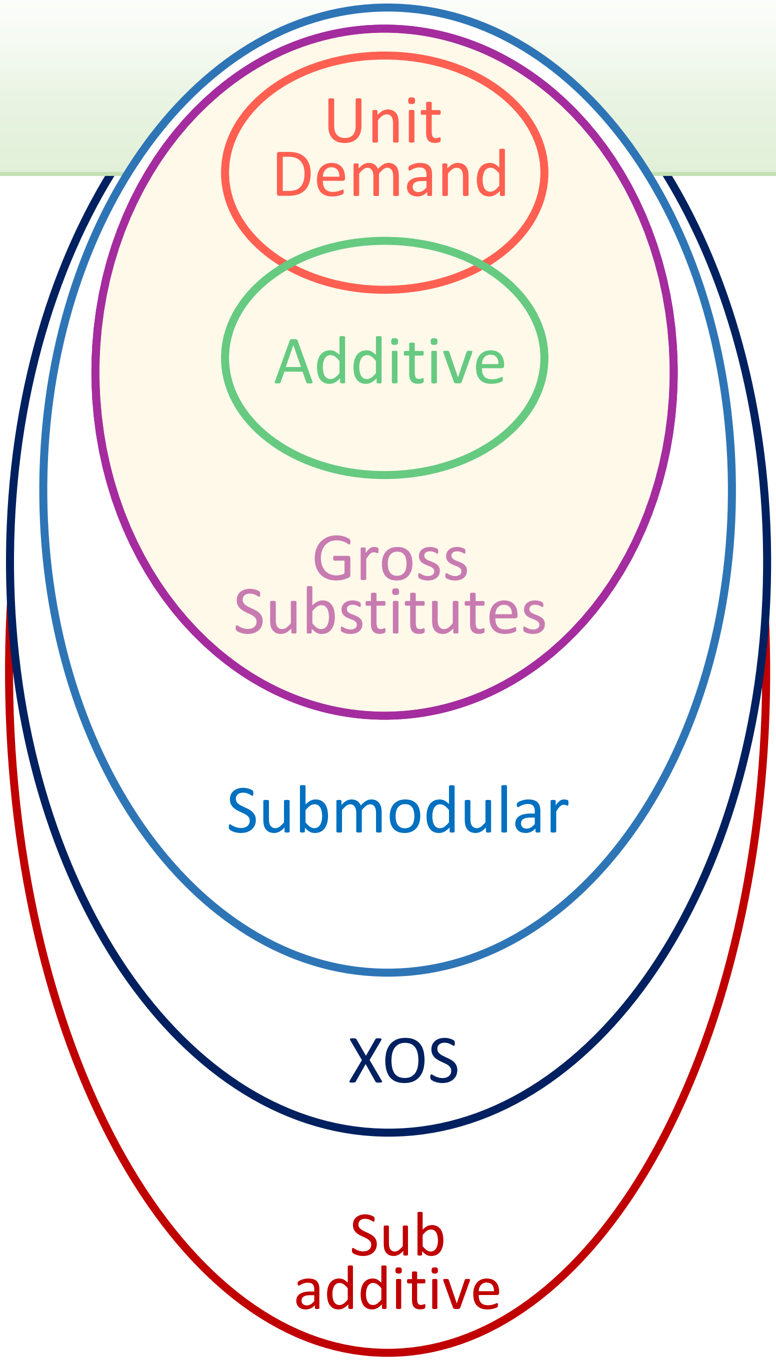
Multiple actions

[Ezra Duetting Feldman Kesselheim FOCS’21]

Single Agent, Many Actions

- n actions $A = \{1, \dots, n\}$, agent chooses a set S
- $c(a) \geq 0$: cost of action a
- $c(S) = \sum_{a \in S} c(a)$ [additive cost]
- $f: 2^A \to [0,1]$ success probability function
  - $f(S)$: success probability for actions $S \subseteq A$
  - Not necessarily additive
- Reward: 1 for success, 0 for failure

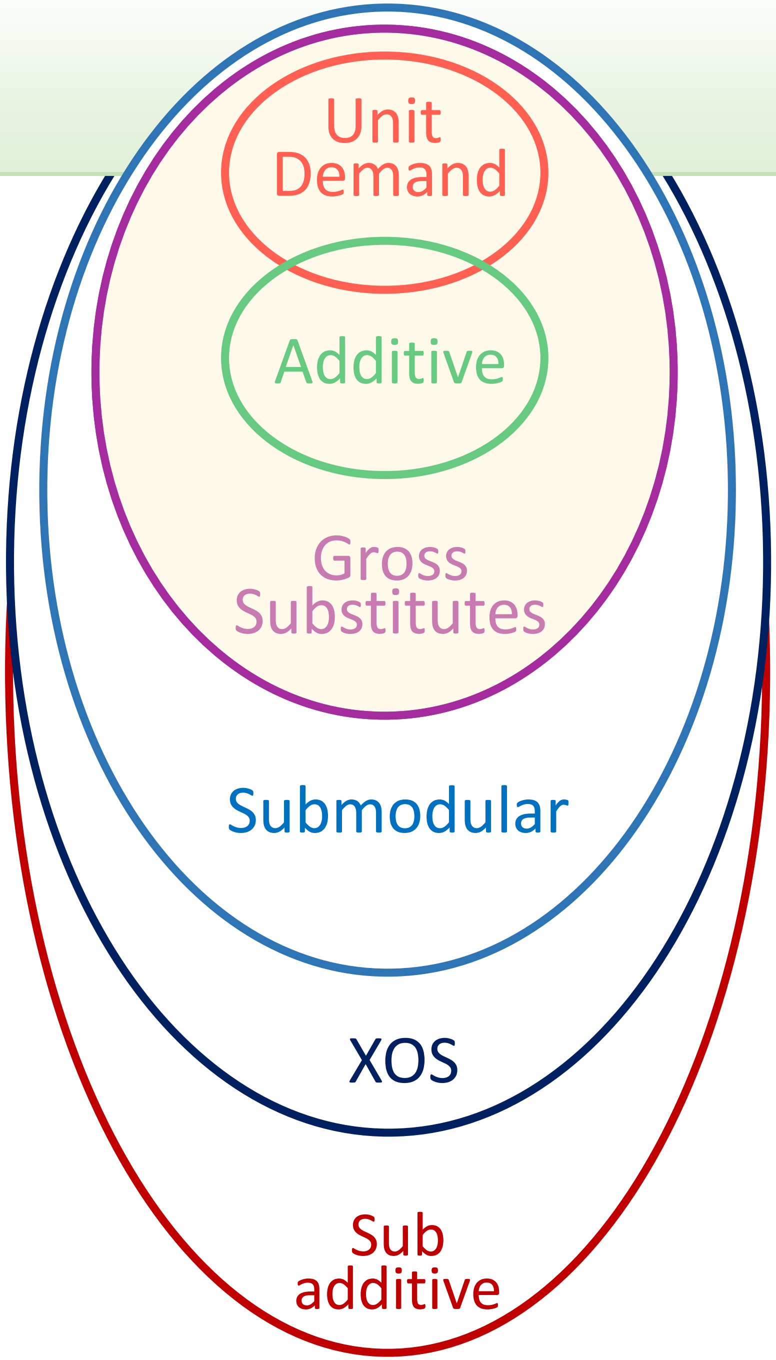
Submodular: $f(j \mid S) \geq f(j \mid T)$ for $S \subseteq T$
(decreasing marginal value)
Subadditive: $f(S) + f(T) \geq f(S \cup T)$



Single Agent, Many Actions

- n actions $A = \{1, \ldots, n\}$, agent chooses a set S
- $c(a) \geq 0$: cost of action a
- $c(S) = \sum_{a \in S} c(a)$ [additive cost]
- $f: 2^A \to [0,1]$ success probability function
  - $f(S)$: success probability for actions $S \subseteq A$
  - Not necessarily additive
- Reward: 1 for success, 0 for failure

Optimal Contract Problem:
Find S and α that maximize $(1-\alpha)f(S)$ [principal's utility]
where S maximizes $\alpha f(S) - c(S)$ [agent's utility]



Main Results

Theorems

- A polynomial-time algorithm for gross substitutes functions
- For submodular functions (i.e., decreasing marginal contribution), it is NP-hard to compute the optimal contract

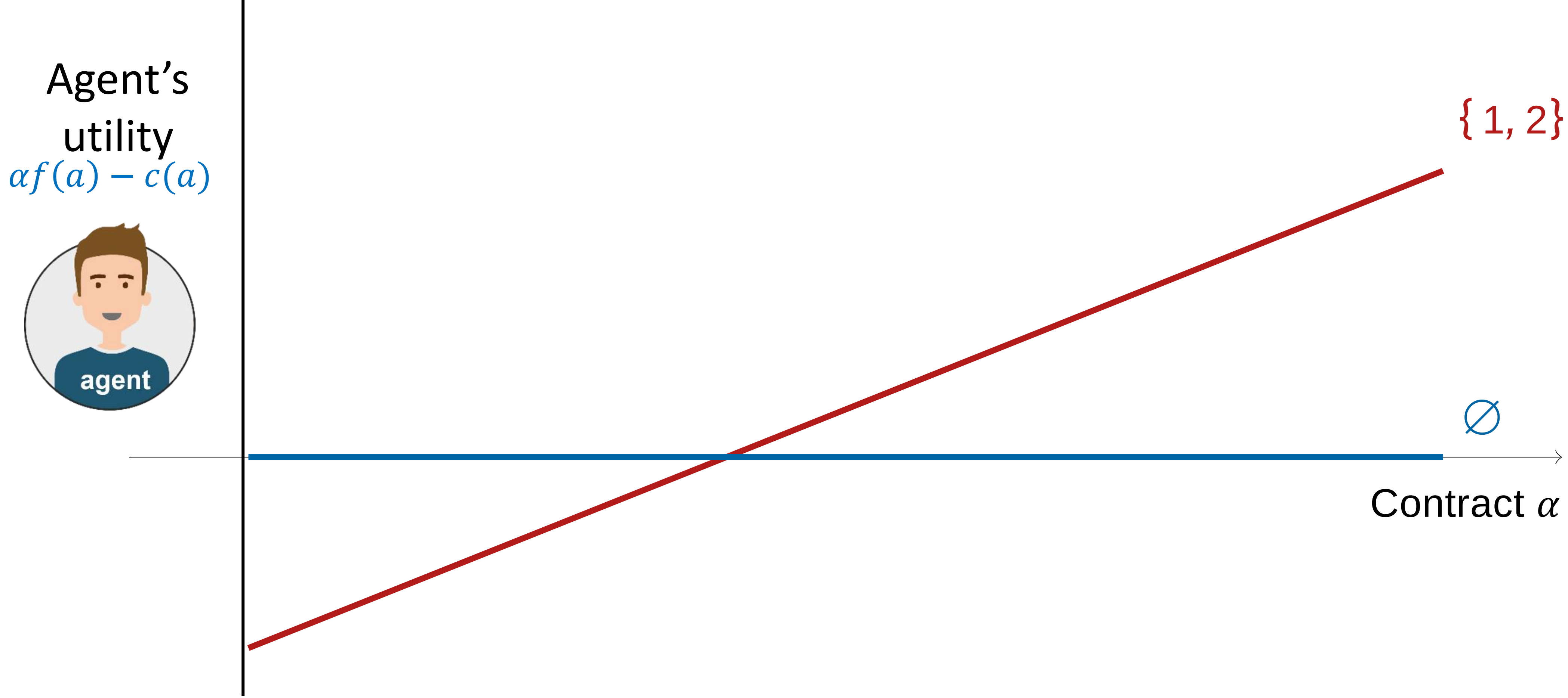
Gross substitutes constitutes a frontier similar to
- welfare maximization tractibility in combinatorial auctions [Nisan Segal 2006]
- market equilibrium existence [Kelso Crawford 1982, Gul Stacchetti 1999]

Upper Envelope: Agent's Perspective

$c(1) = 0.1, c(2) = 0.1, c(3) = 0.4, c(S) = \sum_{a \in S} c(a)$

$f(1) = 0.3, f(2) = 0.2, f(3) = 0.5, f(S) = \sum_{a \in S} f(a)$ (additive)

Upper Envelope: Agent’s Perspective



$c(1) = 0.1, c(2) = 0.1, c(3) = 0.4, c(S) = \sum_{a \in S} c(a)$

$f(1) = 0.3, f(2) = 0.2, f(3) = 0.5, f(S) = \sum_{a \in S} f(a)$ (additive)

Upper Envelope: Agent's Perspective

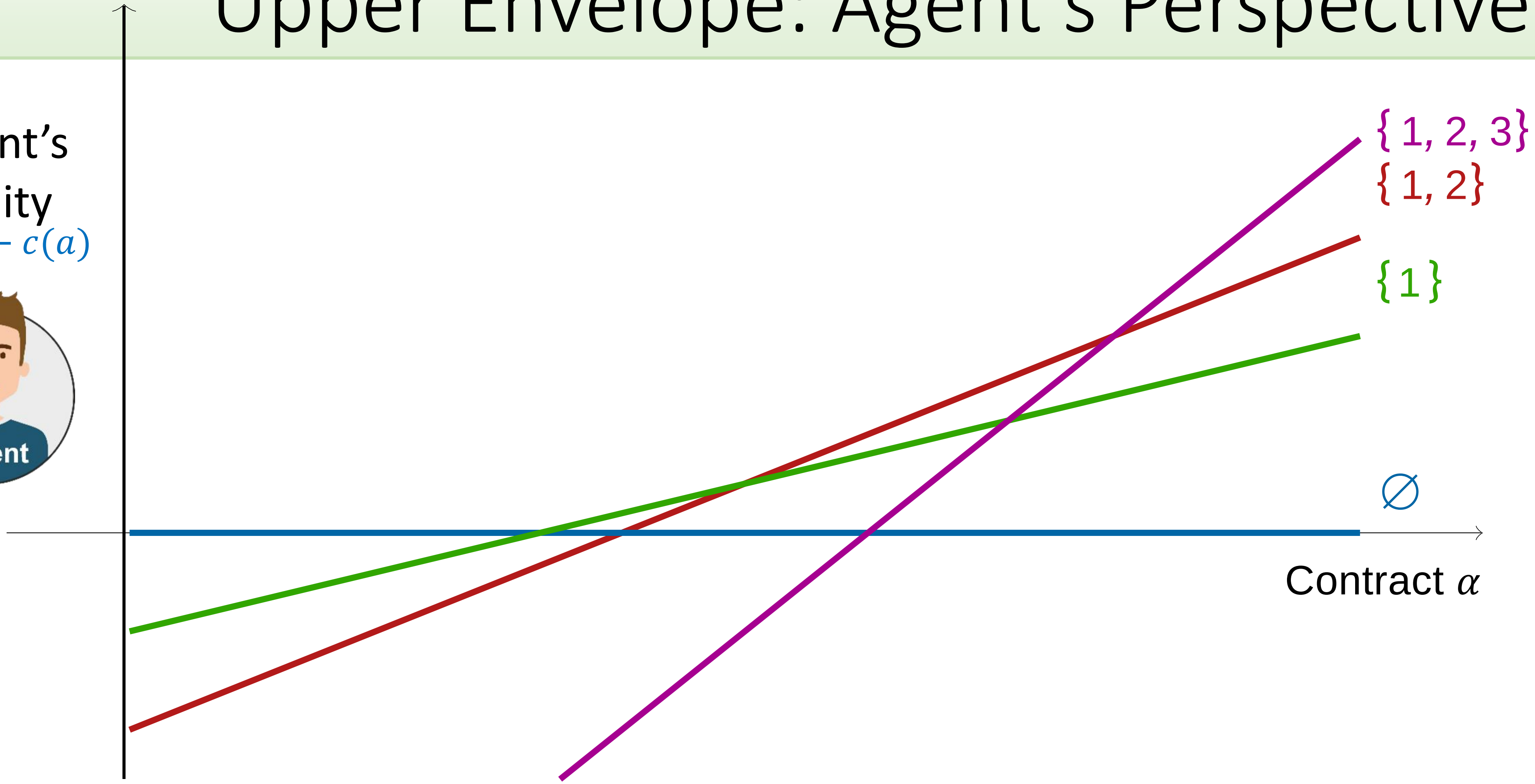
$c(1) = 0.1, c(2) = 0.1, c(3) = 0.4, c(S) = \sum_{a \in S} c(a)$

$f(1) = 0.3, f(2) = 0.2, f(3) = 0.5, f(S) = \sum_{a \in S} f(a)$ (additive)

Upper Envelope: Agent's Perspective

Agent's utility

$\alpha f(a) - c(a)$

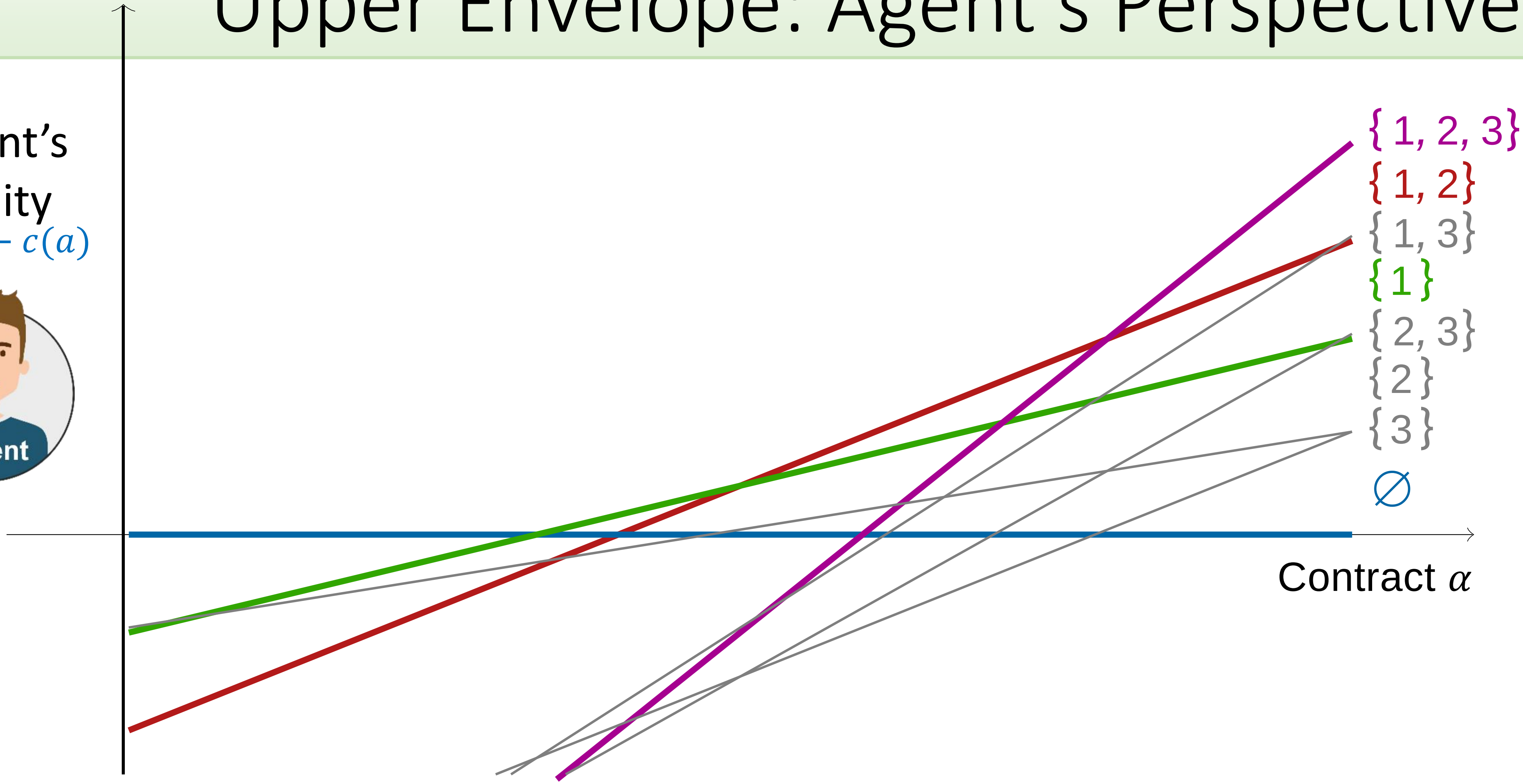


$c(1) = 0.1, c(2) = 0.1, c(3) = 0.4, c(S) = \sum_{a \in S} c(a)$

$f(1) = 0.3, f(2) = 0.2, f(3) = 0.5, f(S) = \sum_{a \in S} f(a)$ (additive)

Upper Envelope: Agent's Perspective

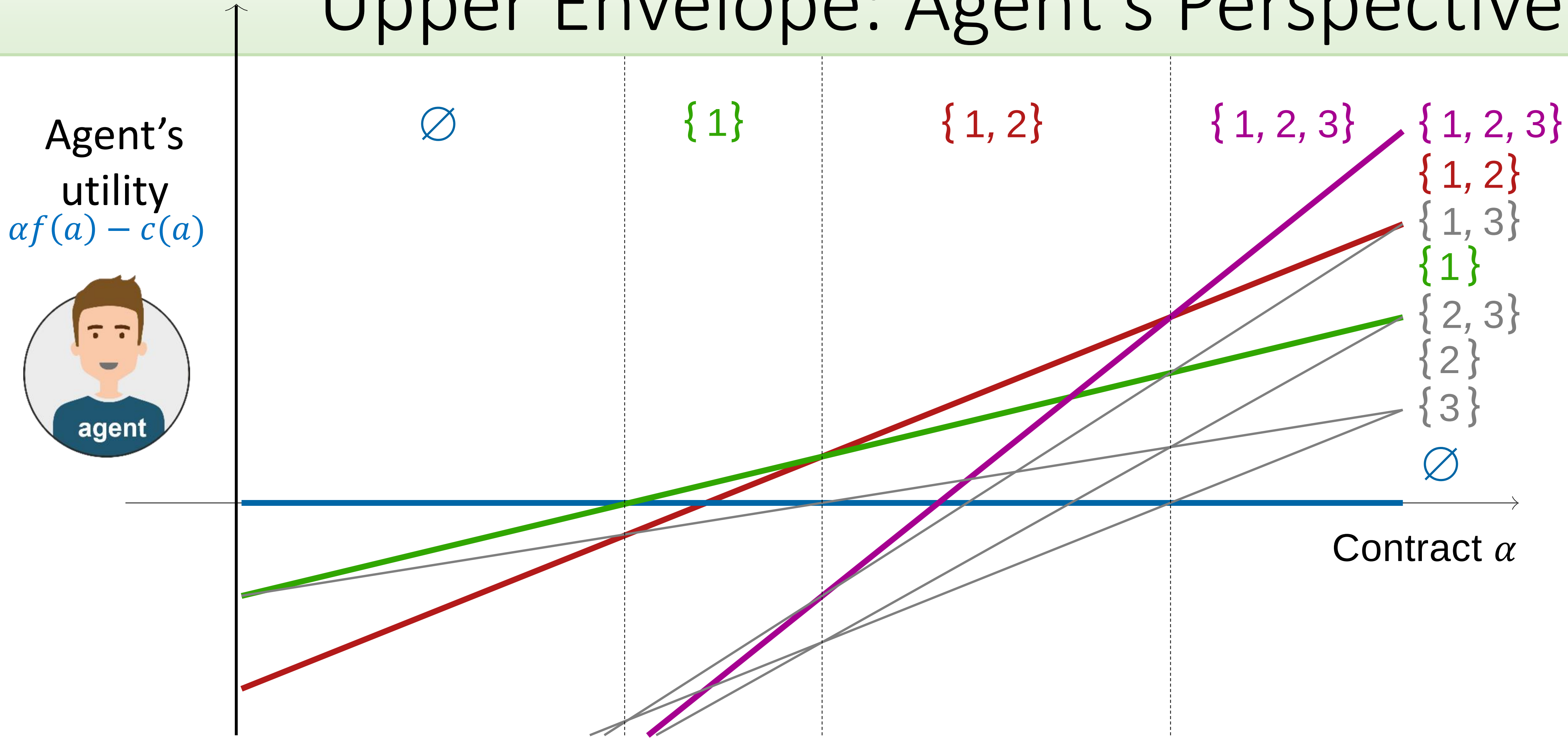
Agent's utility
$\alpha f(a) - c(a)$



$c(1) = 0.1, c(2) = 0.1, c(3) = 0.4, c(S) = \sum_{a \in S} c(a)$

$f(1) = 0.3, f(2) = 0.2, f(3) = 0.5, f(S) = \sum_{a \in S} f(a)$ (additive)

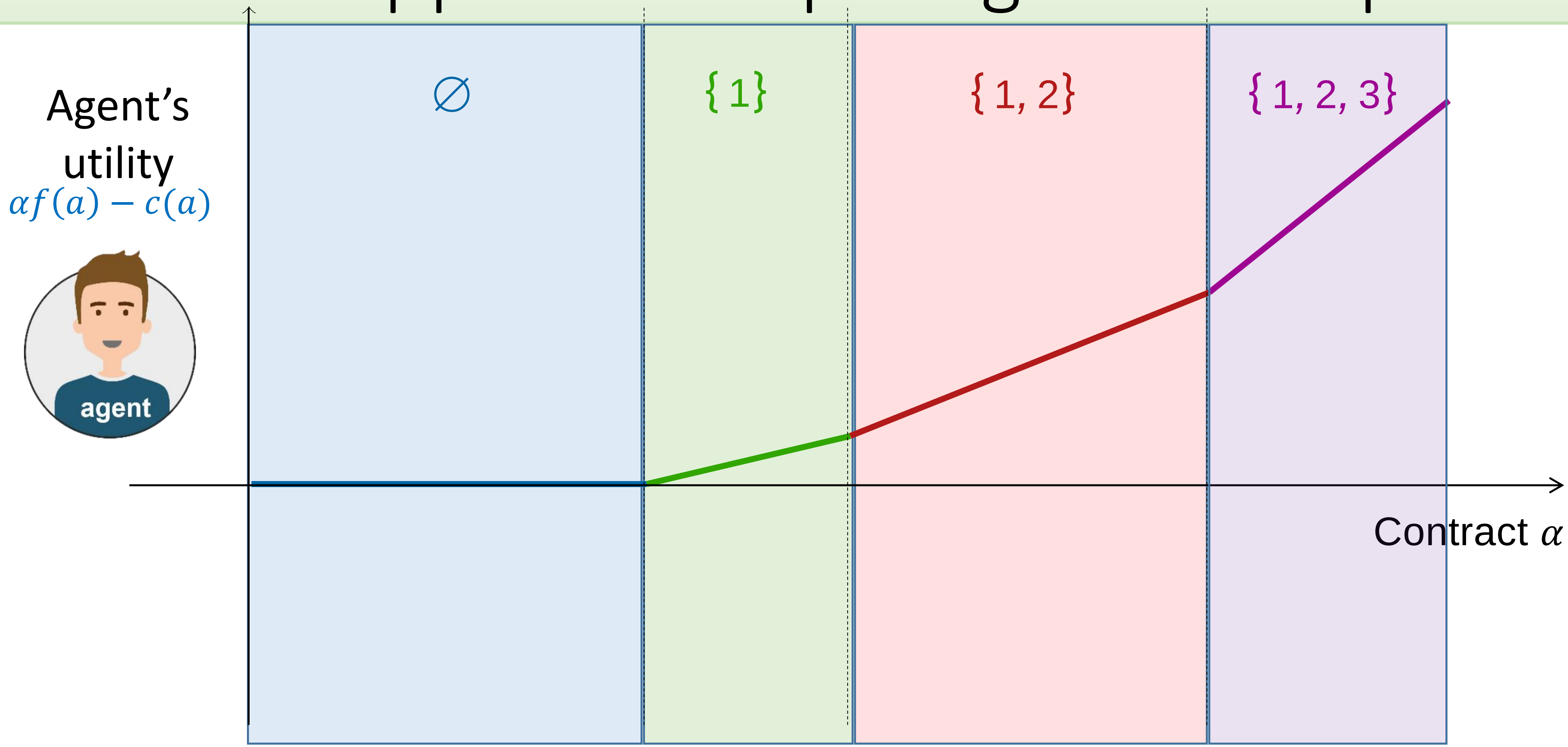
Upper Envelope: Agent's Perspective



$c(1) = 0.1, c(2) = 0.1, c(3) = 0.4, c(S) = \sum_{a \in S} c(a)$

$f(1) = 0.3, f(2) = 0.2, f(3) = 0.5, f(S) = \sum_{a \in S} f(a)$ (additive)

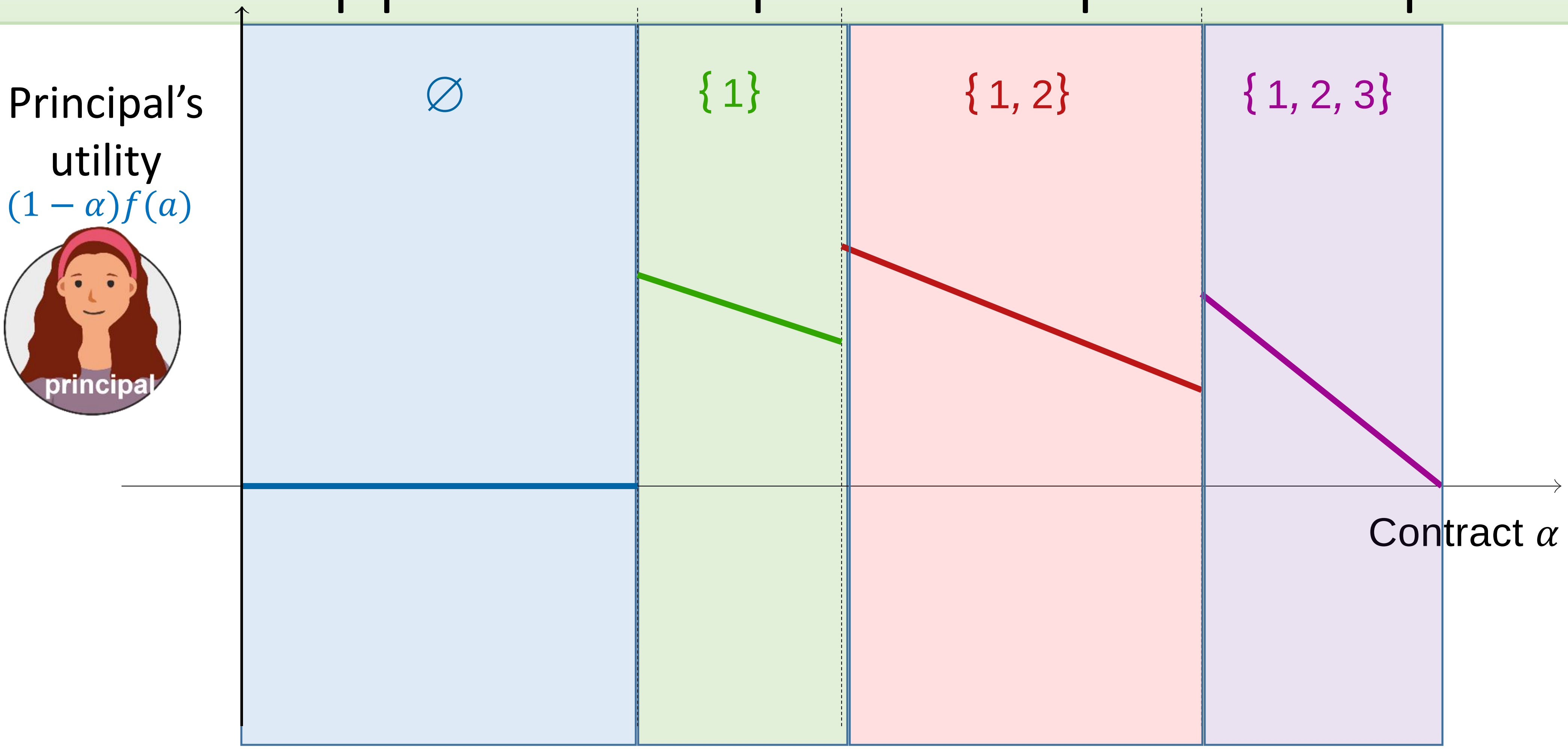
Upper Envelope: Agent's Perspective



$c(1) = 0.1, c(2) = 0.1, c(3) = 0.4, c(S) = \sum_{a \in S} c(a)$

$f(1) = 0.3, f(2) = 0.2, f(3) = 0.5, f(S) = \sum_{a \in S} f(a)$ (additive)

Upper Envelope: Principal's Perspective



$c(1) = 0.1, c(2) = 0.1, c(3) = 0.4, c(S) = \sum_{a \in S} c(a)$

$f(1) = 0.3, f(2) = 0.2, f(3) = 0.5, f(S) = \sum_{a \in S} f(a)$ (additive)

Critical Alphas and an Algorithm

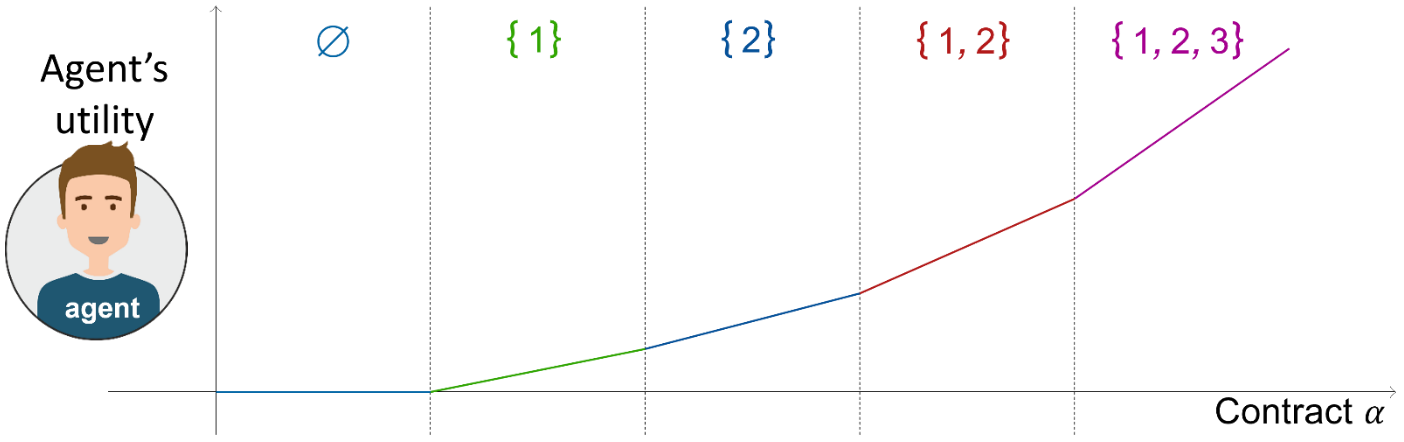
- **Simple observation**: only transition points are interesting
- $C_{f,c}$: Set of critical alphas

  (i.e., points where agent's best response changes)
- Naïve algorithm: Go over all critical alphas and take the best
  - Requires subroutine: find next critical alpha
  - Requires bound on number of critical alphas

Theorem: For gross substitutes, this yields a polynomial-time algorithm

Also: There is a submodular f for which $|C_{f,c}|$ = exponential in n.

Proof Sketch: For GS $|C_{f,c}|$ is polynomial in n



The agent's problem: given α,

find S that maximizes $\alpha f(S) - c(S)$

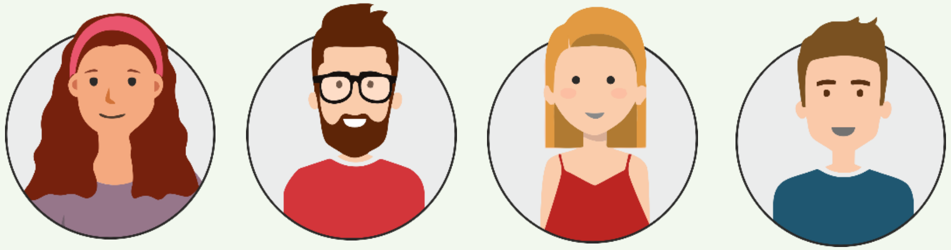
$$\Leftrightarrow$$

find S that maximizes $f(S) - \frac{1}{\alpha} c(S)$

- This is precisely a demand query!
- **Non-standard:** All prices go down simultaneously, at rate $\frac{1}{\alpha}$
- **Theorem:** For GS functions, at each critical point:
  - an action is added to S, or
  - an action from S is replaced by one with higher cost
- Potential function argument showing that $|C_{f,c}| = O(n^2)$

Multiple Actions: Summary

- **Key take-aways:**
  - Gross substitutes also ``frontier of tractability'' for combinatorial contracts
  - (Perhaps) surprising connection to auctions
- **Additional results in the paper:**
  - FPTAS for general functions f
  - Robust optimality of linear contracts for non-binary outcomes
  - Extension of computational results to linear contracts for non-binary outcomes
- **Many open problems:**
  - Polynomial-time algorithm for submodular valuations with demand queries?
  - Extension to multiple agents



Multiple agents

[Feldman, Chuang, Stoica, Shenker EC'05, Babaioff Feldman Nisan EC'06, Emek Feldman '09, Ezra Duetting Feldman Kesselheim, working paper]

Combinatorial Agency Model

[Babaioff, Feldman, Nisan 2006]

- n agents
- Binary action: $A_i = \{0,1\}$
  (0: no effort, 1: effort)
- Cost c_i: cost of effort (no effort = no cost)
- Binary outcome: {0,1}
- Principal receives reward 1 for success
- Success probability function $f: \{0,1\}^n \rightarrow [0,1]$

Contracts and Objective

- Optimal (=linear) contract: $\alpha = (\alpha_1, \dots, \alpha_n)$
  $\alpha_i \geq 0$: payment to agent i for success
- "margin" of i w.r.t. S: $f(i \mid S - i) = f(S) - f(S - i)$
- **Agent's perspective:** Agent i prefers ``effort'' over ``no effort'' iff

$$\alpha_i f(S) - c_i \geq \alpha_i f(S - \{i\})$$

$\Rightarrow \quad \alpha_i = \frac{c_i}{f(i \mid S - i)}$ is the best way to incentivize agent i

- **Principal's perspective:** Find the set of agents S that maximizes

$$g(S) = f(S)\left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S - i)}\right)$$

- **Problem**: compute optimal contract for submodular/XOS/subadditive f
- **Challenge**: even if f is highly structured, g is a mess

Warmup: Additive f

Theorem: The problem is NP-hard even for additive f, but admits an FPTAS

Proof: via reduction from PARTITION

- PARTITION: given a multiset of integers that sum to W, determine whether one can partition them into to sets that sum to $W/2$
- Construct a contract instance (i.e., $\{f_i\}, \{c_i\}$) where the principal's utility is maximized when the sum of agent values sum to $W/2$

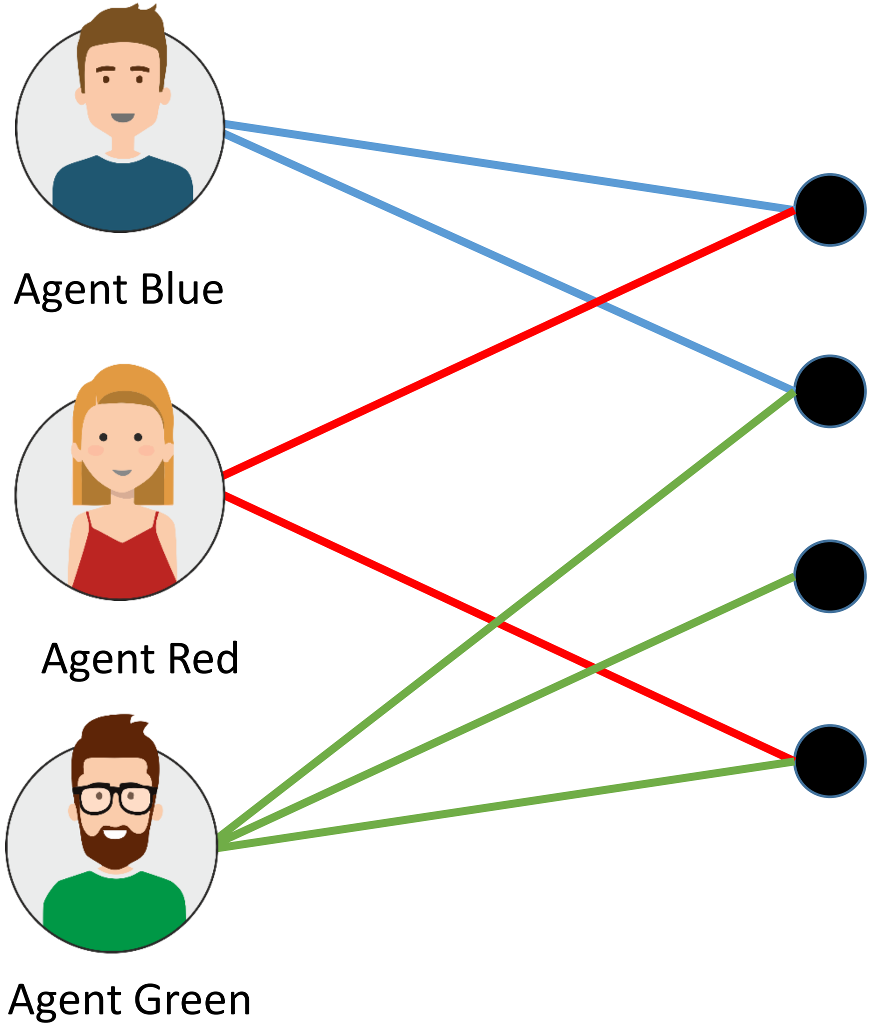
Next: Submodular/XOS/Subadditive f

Submodular: $f(i \mid S) \geq f(i \mid T)$ for $S \subseteq T$ (decreasing marginal value)

XOS: maximum over additive

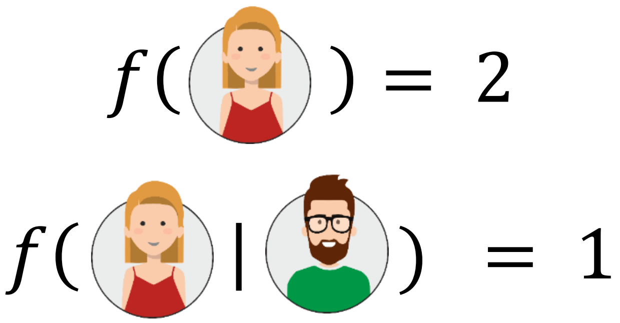
Subadditive: $f(S) + f(T) \geq f(S \cup T)$

Unweighted Coverage Function (submodular)

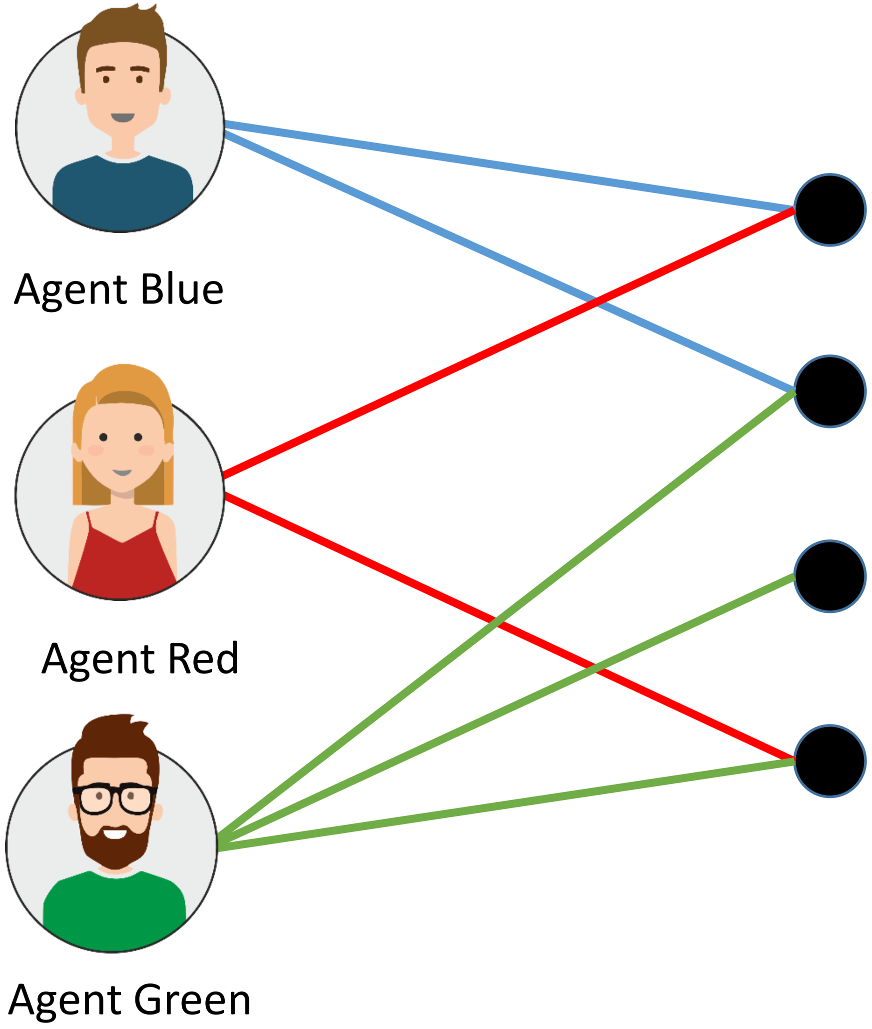


f(set of agents) $=$
tasks covered by these agents

e.g.:



Unweighted Coverage Function (submodular)



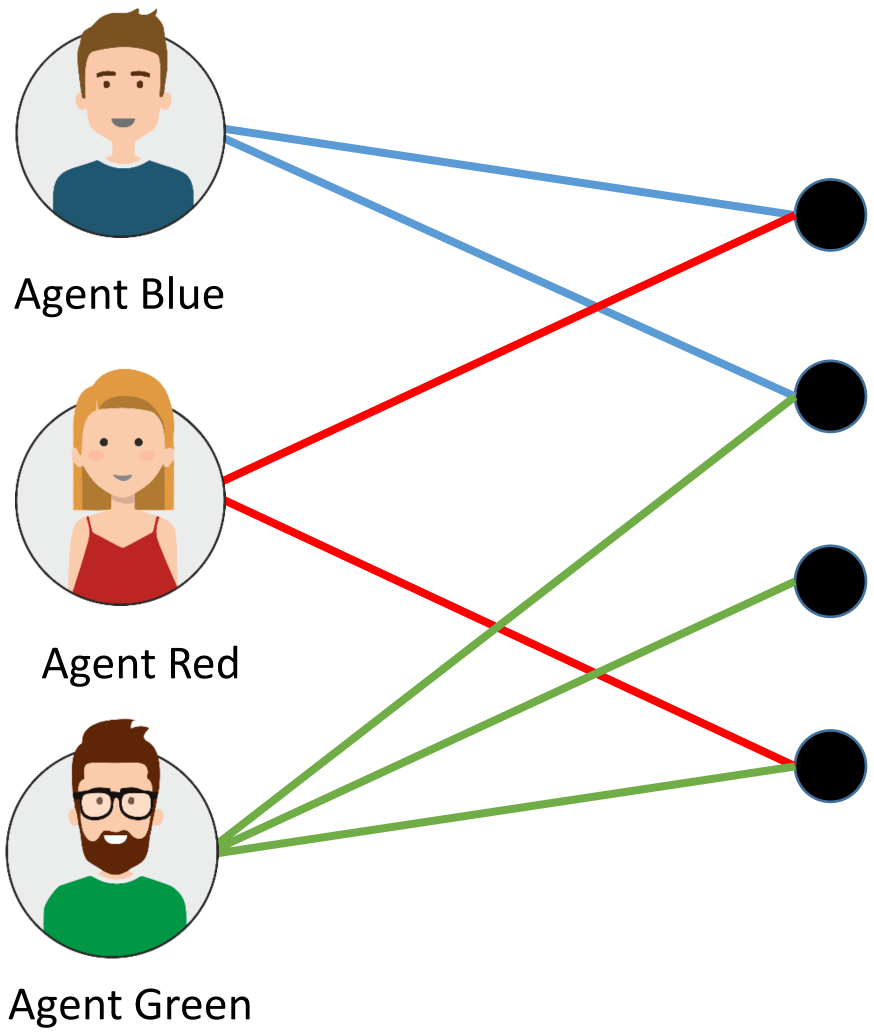
Principal's objective:

$$g(S) = f(S)\left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S - i)}\right)$$

Total # tasks covered by S

tasks covered uniquely by agent i

Unweighted Coverage Function (submodular)



Principal's objective:

$$g(S) = f(S)\left(1 - \sum_{i \in S} \frac{c_i}{f(i\,|S - i)}\right)$$

Total # tasks covered by S

tasks covered uniquely by agent i

Unique coverage is hard to approximate within a constant factor [Demaine Feige Hajiaghayi Salavatipour 2006]

Approximation Results for Submodular/XOS/Subadditive

[Dutting Ezra Feldman Kesselheim, Working Paper]

Given action "prices" $p_1, \ldots, p_n$, return S maximizing $f(S) - \sum_{i \in S} p_i$

Given S, return $f(S)$

Results:

- **(+)** There is a polynomial-time algorithm for finding a $O(1)$-approximate contract for submodular f, using value oracle, and for XOS f, using demand oracle

Approximation Results for Submodular/XOS/Subadditive

[Dutting Ezra Feldman Kesselheim, Working Paper]

Given action "prices" $p_1, \ldots, p_n$, return S maximizing $f(S) - \sum_{i \in S} p_i$

Given S, return $f(S)$

Results:

- **(+)** There is a polynomial-time algorithm for finding a $O(1)$-approximate contract for submodular f, using value oracle, and for XOS f, using demand oracle
- **(-)** No better than constant-approximation for XOS f, using demand and value oracles
- **(-)** No better than $\Omega(\sqrt{n})$-approximation for subadditive f, using demand and value oracles (even for f constant close to submodular)

$$g(S) = f(S)(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S_{-i})})$$

Proof Sketch (XOS)

- **Goal**: find S satisfying $g(S) \geq \text{const} \cdot g(S^*)$, where S^* is optimal set
- Let T be the demand set under prices $p_i = \frac{1}{2}\sqrt{c_i f(S^*)}$
- **Lemma 1**: $f(T) \geq \frac{1}{2} f(S^*)$ [so we can get a set that approximates $f(S^*)$]
- **Lemma 2**: For every set S, if $f(i \mid S - \mathrm{i}) \geq \sqrt{2c_i f(S)}$ for all $i \in S$, then $g(S) \geq \frac{1}{2} f(S)$ (so, sufficient to approximate f, instead of messy g)
- Since T is a demand set, $f(i \mid T - i) \geq p_i = \frac{1}{2}\sqrt{c_i f(S^*)}$

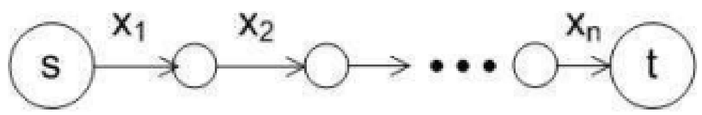
$g(S) = f(S)(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S_{-i})})$

Proof Sketch (XOS)

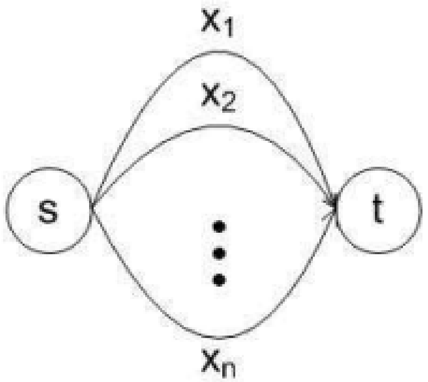
- **Goal**: find S satisfying $g(S) \geq \text{const} \cdot g(S^*)$, where S^* is optimal set
- Let T be the demand set under prices $p_i = \frac{1}{2}\sqrt{c_i f(S^*)}$
- **Lemma 1**: $f(T) \geq \frac{1}{2} f(S^*)$ [so we can get a set that approximates $f(S^*)$]
- **Lemma 2**: For every set S, if $f(i \mid S - \mathrm{i}) \geq \sqrt{2c_i f(S)}$ for all $i \in S$, then $g(S) \geq \frac{1}{2} f(S)$ (so, sufficient to approximate f, instead of messy g)
- Since T is a demand set, $f(i \mid T - i) \geq p_i = \frac{1}{2}\sqrt{c_i f(S^*)} \geq \sqrt{2c_i f(T)}$ ← We wish that (to use Lemma 2)
  - **Problem**: $f(T)$ may be too large
  - **Idea**: remove items from T until inequality is satisfied
  - **Problem**: marginals may decrease (unlike submodular)
- **Thm**: a novel scaling property of XOS: scale down $f(T)$ and keep marginals high enough
- **Altogether**: $g(S) \geq \frac{1}{2} f(S) \geq \text{const} \cdot f(T) \geq \text{const} \cdot f(S^*) \geq \text{const} \cdot g(S^*)$ ∎

How this Fits into Known Results

- Babaioff Feldman Nisan (2006):
  - For general f: exp. many value queries
  - For f encoded as read-once network: #P-complete
  - Poly-time algorithm for AND read-once networks
  - Conjecture: Polynomial-time algorithm for series-parallel read-once networks
- Feldman and Emek (2009):
  - NP-hard + FPTAS for OR read-once network
  - "Almost FPTAS" for series-parallel read once networks



(a) AND technology



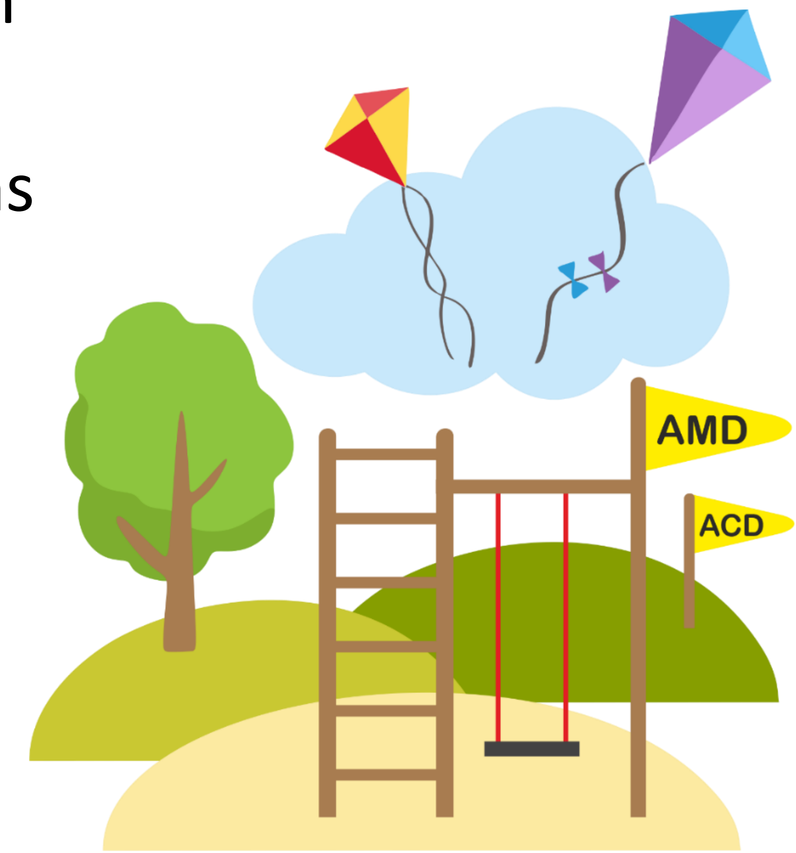
(b) OR technology

Summary

- **Key take-aways:**
  - First constant-factor approximation for a contract problem
  - New properties of XOS functions, that may be of independent interest
- **Many open problems:**
  - Beyond binary action?
  - Approximation alg for general series-parallel graphs?

Main Take Aways

- Contract theory is a new frontier in AGT
- Complexity and approximation shed new light on contract design
- Interesting connections to combinatorial auctions and other combinatorial optimization problems
  - E.g., gross substitutes as tractability frontier
- Many fundamental problems still open



Thank You!