LECTURE 7: Reaching Agreements

An Introduction to MultiAgent Systems
http://www.csc.liv.ac.uk/~mjw/pubs/imas

Reaching Agreements

- How do agents *reaching agreements* when they are self interested?
- In an extreme case (zero sum encounter) no agreement is possible — but in most scenarios, there is potential for *mutually beneficial agreement* on matters of common interest
- The capabilities of *negotiation* and *argumentation* are central to the ability of an agent to reach such agreements

Mechanisms, Protocols, and Strategies

- Negotiation is governed by a particular *mechanism*, or *protocol*
- The mechanism defines the “rules of encounter” between agents
- *Mechanism design* is designing mechanisms so that they have certain desirable properties
- Given a particular protocol, how can a particular *strategy* be designed that individual agents can use?

Mechanism Design

- Desirable properties of mechanisms:
  - *Convergence/guaranteed success*
  - *Maximizing social welfare*
  - *Pareto efficiency*
  - *Individual rationality*
  - *Stability*
  - *Simplicity*
  - *Distribution*

Auctions

- An auction takes place between an agent known as the *auctioneer* and a collection of agents known as the *bidders*
- The goal of the auction is for the auctioneer to allocate the *good* to one of the bidders
- In most settings the auctioneer desires to maximize the price; bidders desire to minimize price

Auction Parameters

- *Goods can have*
  - *private value*
  - *public/common value*
  - *correlated value*
- *Winner determination may be*
  - *first price*
  - *second price*
- *Bids may be*
  - *open cry*
  - *sealed bid*
- *Bidding may be*
  - *one shot*
  - *ascending*
  - *descending*

English Auctions

- Most commonly known type of auction:
  - *first price*
  - *open cry*
  - *ascending*
- Dominant strategy is for agent to successively bid a small amount more than the current highest bid until it reaches their valuation, then withdraw
- Susceptible to:
  - *winner’s curse*
  - *shills*

Dutch Auctions

- Dutch auctions are examples of *open-cry descending* auctions:
  - auctioneer starts by offering good at artificially high value
  - auctioneer lowers offer price until some agent makes a bid equal to the current offer price
  - the good is then allocated to the agent that made the offer

First-Price Sealed-Bid Auctions

- First-price sealed-bid auctions are *one-shot auctions*:
  - there is a single round
  - bidders submit a sealed bid for the good
  - good is allocated to agent that made highest bid
  - winner pays price of highest bid
- Best strategy is to *bid less than true valuation*

Vickrey Auctions

- Vickrey auctions are:
  - *second-price*
  - *sealed-bid*
- Good is awarded to the agent that made the highest bid; at the price of the *second highest* bid
- *Bidding to your true valuation is dominant strategy in Vickrey auctions*
- Vickrey auctions susceptible to *antisocial* behavior

Lies and Collusion

- The various auction protocols are susceptible to lying on the part of the auctioneer, and collusion among bidders, to varying degrees
- All four auctions (English, Dutch, First-Price Sealed Bid, Vickrey) can be manipulated by bidder collusion
- A dishonest auctioneer can exploit the Vickrey auction by lying about the 2nd-highest bid
- *Shills* can be introduced to inflate bidding prices in English auctions

Negotiation

- Auctions are *only* concerned with the allocation of goods: richer techniques for reaching agreements are required
- *Negotiation* is the process of reaching agreements on matters of common interest
- Any negotiation setting will have four components:
  - A negotiation set: possible proposals that agents can make
  - A protocol
  - Strategies, one for each agent, which are private
  - A rule that determines when a deal has been struck and what the agreement deal is
- Negotiation usually proceeds in a series of rounds, with every agent making a proposal at every round

Negotiation in Task-Oriented Domains

Imagine that you have three children, each of whom needs to be delivered to a different school each morning. Your neighbor has four children, and also needs to take them to school. Delivery of each child can be modeled as an indivisible task. You and your neighbor can discuss the situation, and come to an agreement that it is better for both of you (for example, by carrying the other's child to a shared destination, saving him the trip). There is no concern about being able to achieve your task by yourself. The worst that can happen is that you and your neighbor won't come to an agreement about setting up a car pool, in which case you are no worse off than if you were alone. You can only benefit (or do no worse) from your neighbor's tasks. Assume, though, that one of my children and one of my neighbors' children both go to the same school (that is, the cost of carrying out these two deliveries, or two tasks, is the same as the cost of carrying out one of them). It obviously makes sense for both children to be taken together, and only my neighbor or I will need to make the trip to carry out both tasks.

--- *Rules of Encounter*, Rosenschein and Zlotkin, 1994

Machines Controlling and Sharing Resources

- Electrical grids (load balancing)
- Telecommunications networks (routing)
- PDA’s (schedulers)
- Shared databases (intelligent access)
- Traffic control (coordination)

Heterogeneous, Self-motivated Agents

The systems:

- are not centrally designed
- do not have a notion of global utility
- are dynamic (e.g., new types of agents)
- will not act “benevolently” unless it is in their interest to do so

The Aim of the Research

- Social engineering for communities of machines
  - The creation of interaction environments that foster certain kinds of social behavior

The exploitation of game theory tools for high-level protocol design

Broad Working Assumption

- Designers (from different companies, countries, etc.) come together to agree on *standards* for how their automated agents will interact (in a given domain)
- Discuss various possibilities and their tradeoffs, and agree on protocols, strategies, and social laws to be implemented in their machines

Attributes of Standards

✓	*Efficient*:	Pareto Optimal
✓	*Stable*:	No incentive to deviate
✓	*Simple*:	Low computational and communication cost
✓	*Distributed*:	No central decision-maker
✓	*Symmetric*:	Agents play equivalent roles

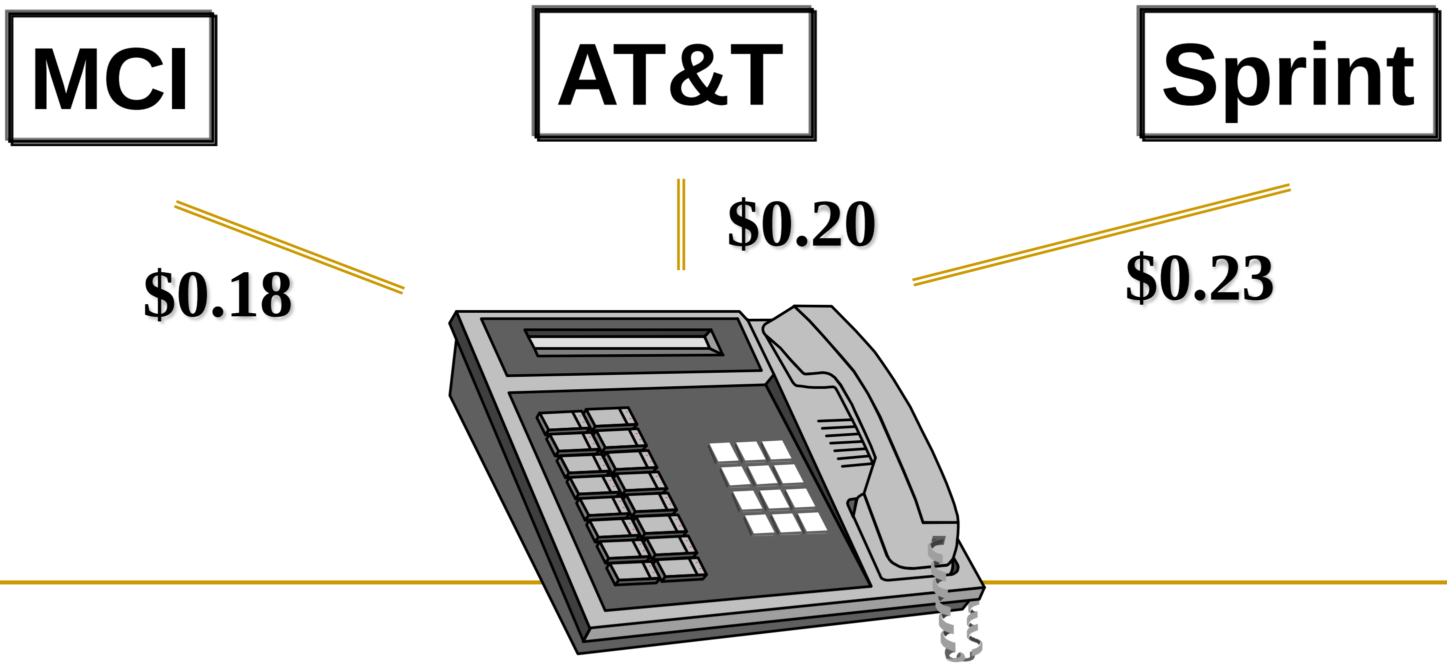
Designing protocols for specific classes of domains that satisfy some or all of these attributes

Distributed Artificial Intelligence (DAI)

- Distributed Problem Solving (DPS)
  - Centrally designed systems, built-in cooperation, have *global* problem to solve
- ✓ Multi-Agent Systems (MAS)
  - ✓ Group of utility-maximizing heterogeneous agents co-existing in same environment, possibly competitive

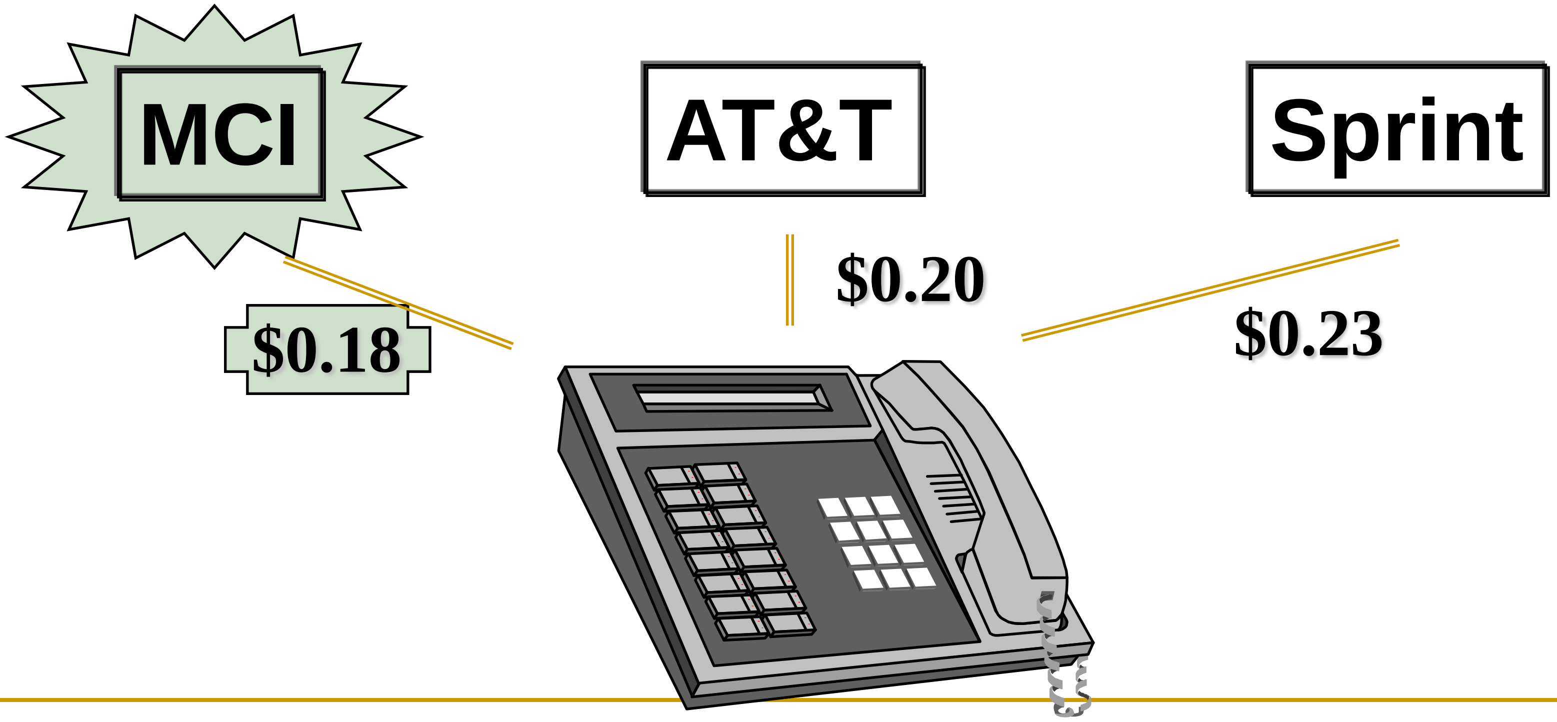
Phone Call Competition Example

- Customer wishes to place long-distance call
- Carriers simultaneously bid, sending proposed prices
- Phone automatically chooses the carrier (dynamically)



Best Bid Wins

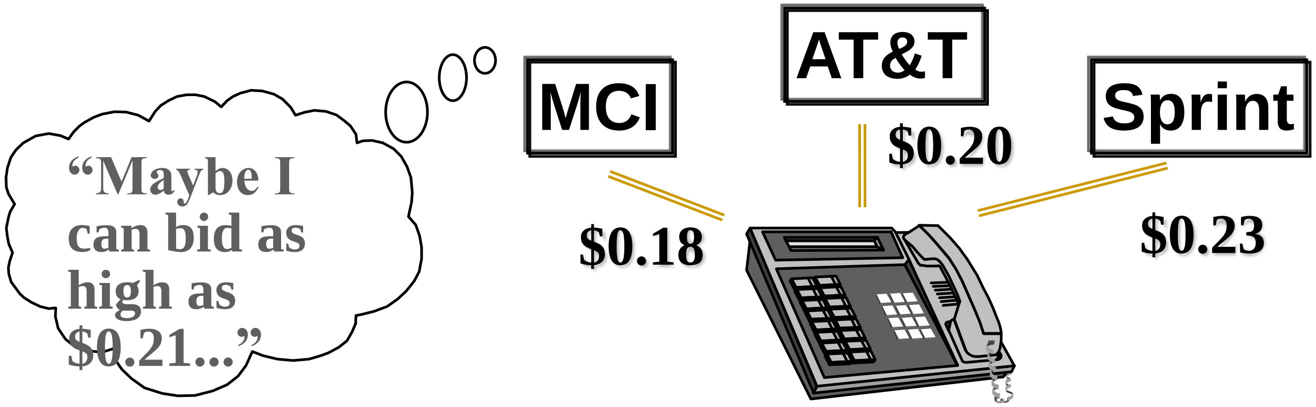
- Phone chooses carrier with lowest bid
- Carrier gets amount that it bid



Attributes of the Mechanism

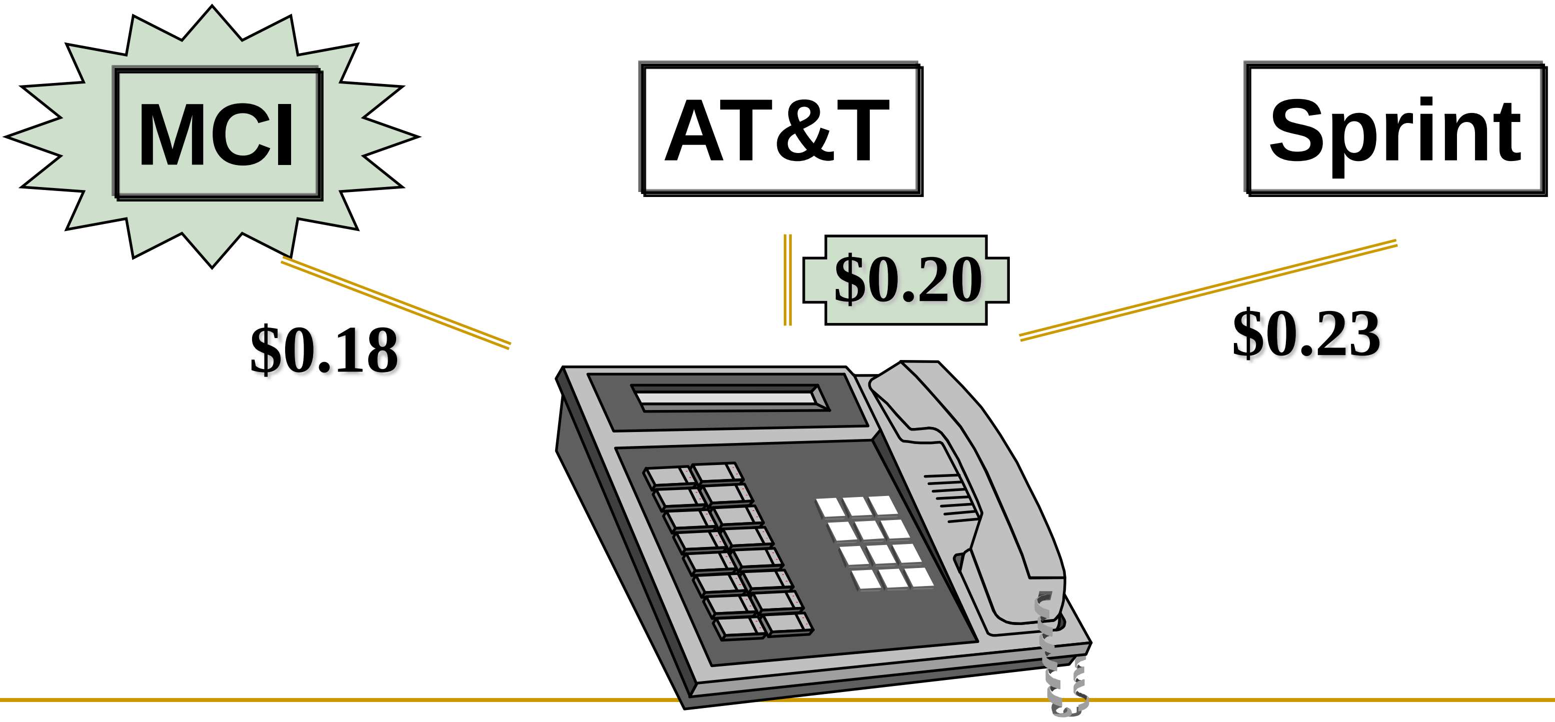
- ✓ *Distributed*
- ✓ *Symmetric*
- × *Stable*
- × *Simple*
- × *Efficient*

Carriers have an incentive to invest effort in strategic behavior



Best Bid Wins, Gets Second Price (Vickrey Auction)

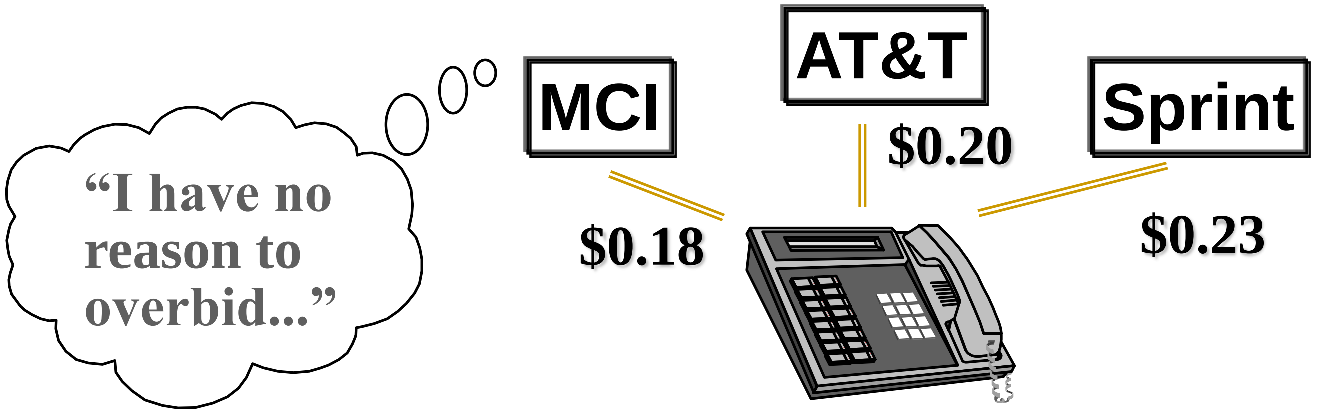
- Phone chooses carrier with lowest bid
- Carrier gets amount of second-best price



Attributes of the Vickrey Mechanism

- ✓ *Distributed*
- ✓ *Symmetric*
- ✓ *Stable*
- ✓ *Simple*
- ✓ *Efficient*

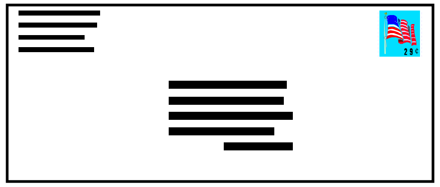
Carriers have *no* incentive to invest effort in strategic behavior



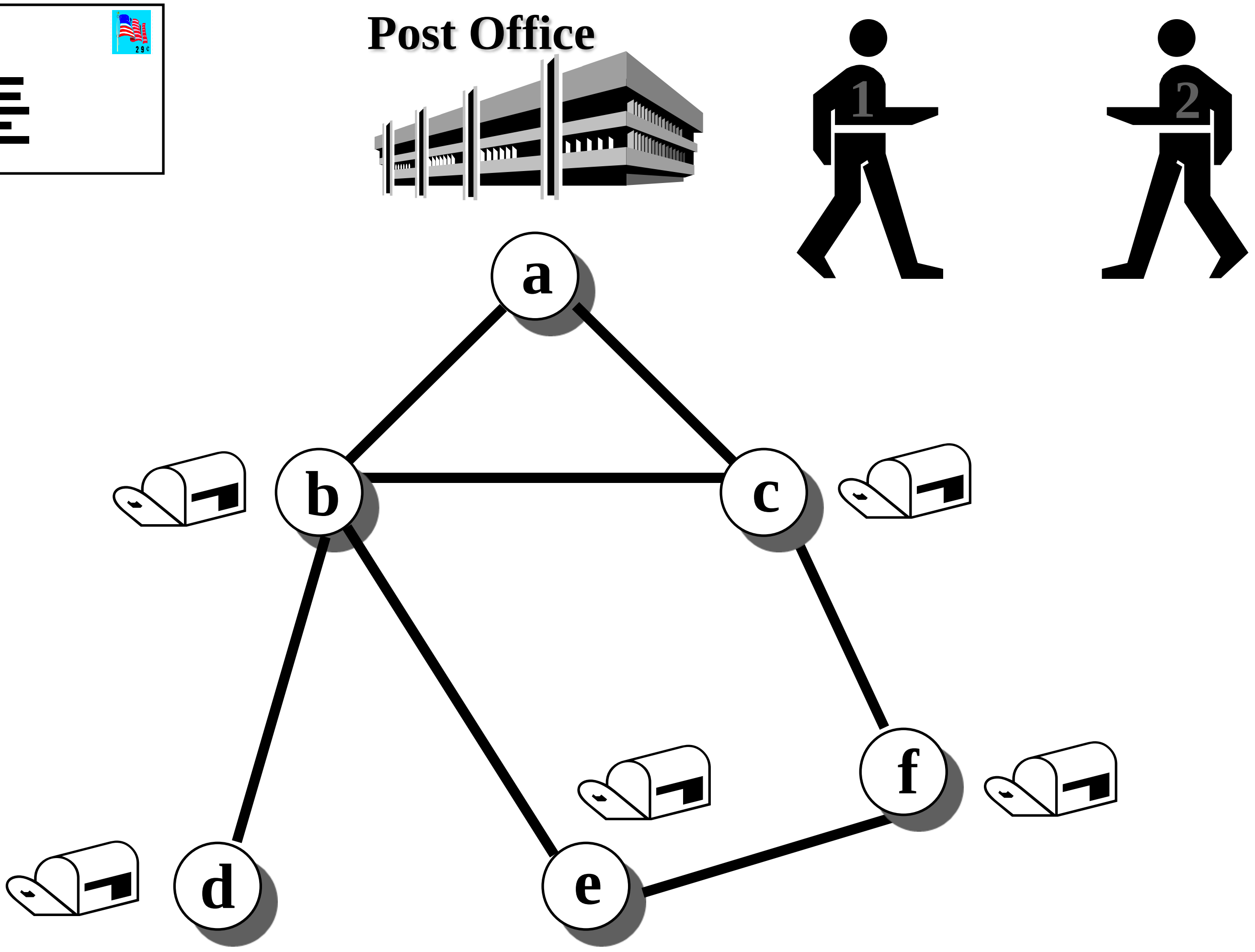
Domain Theory

- Task Oriented Domains
  - Agents have tasks to achieve
  - → Task redistribution
- State Oriented Domains
  - Goals specify acceptable final states
  - Side effects
  - → Joint plan and schedules
- Worth Oriented Domains
  - Function rating states' acceptability
  - → Joint plan, schedules, and goal relaxation

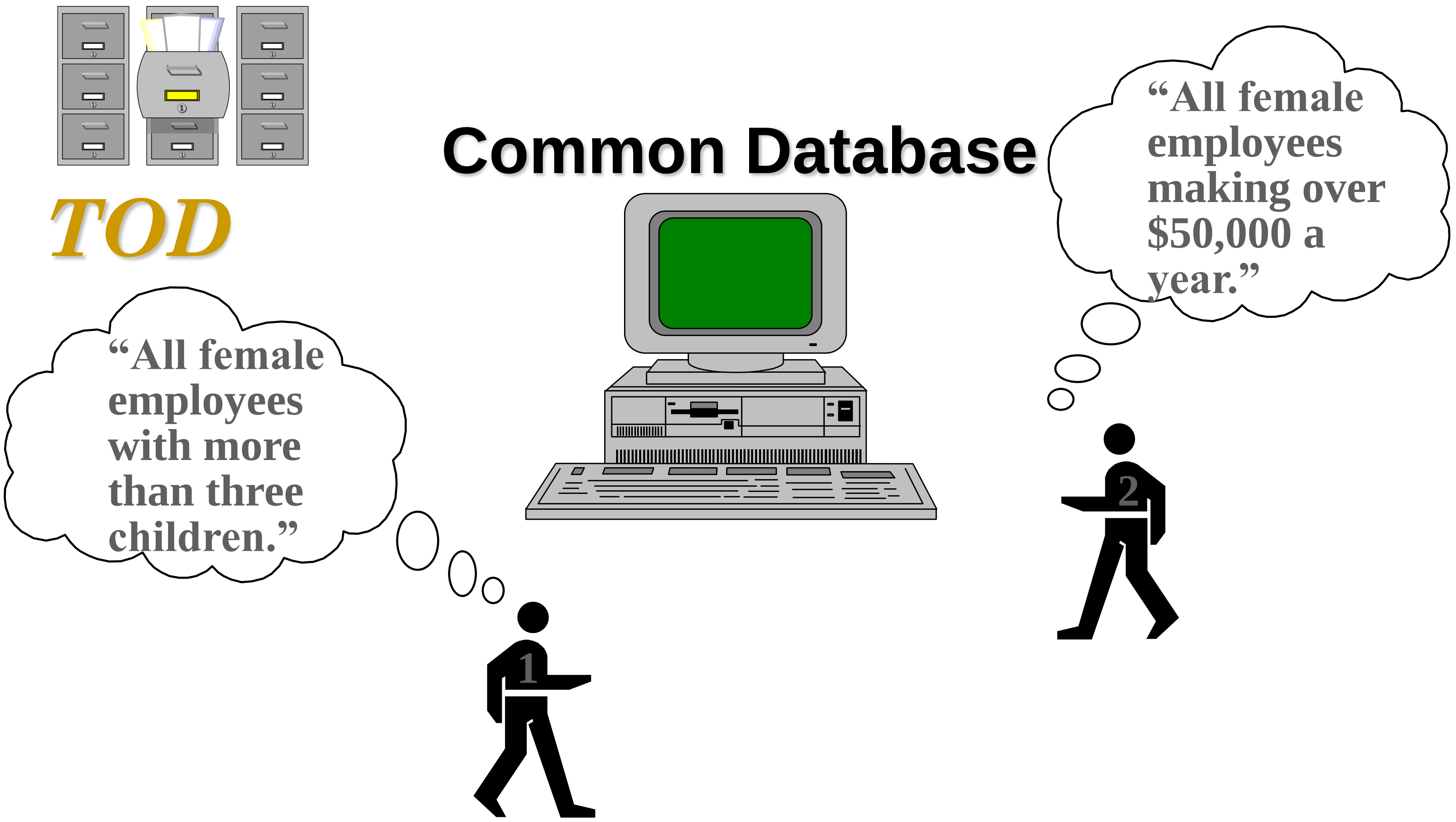
Postmen Domain



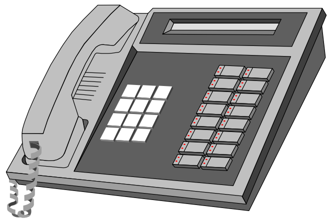
TOD



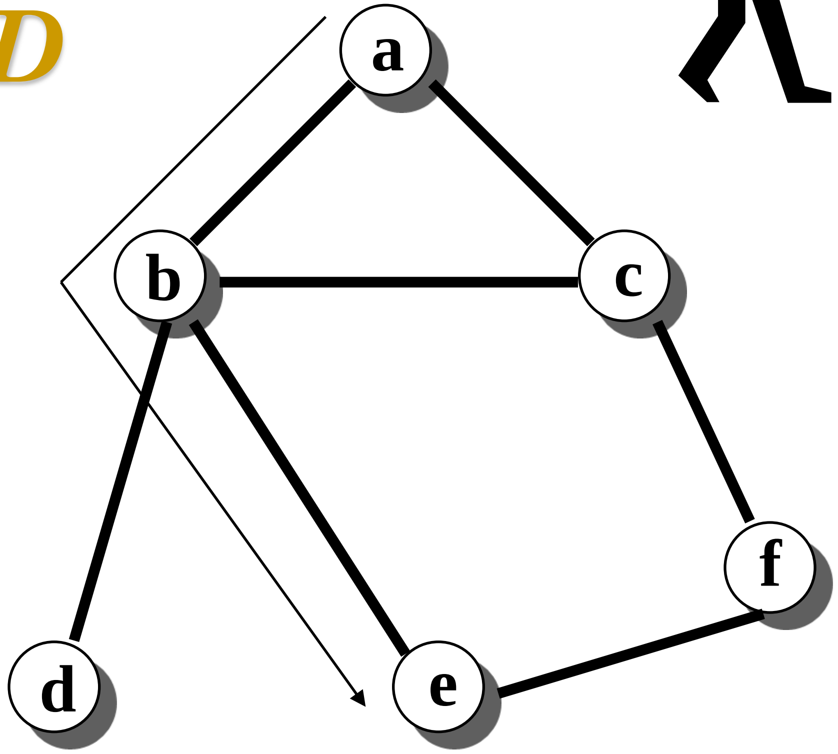
Database Domain

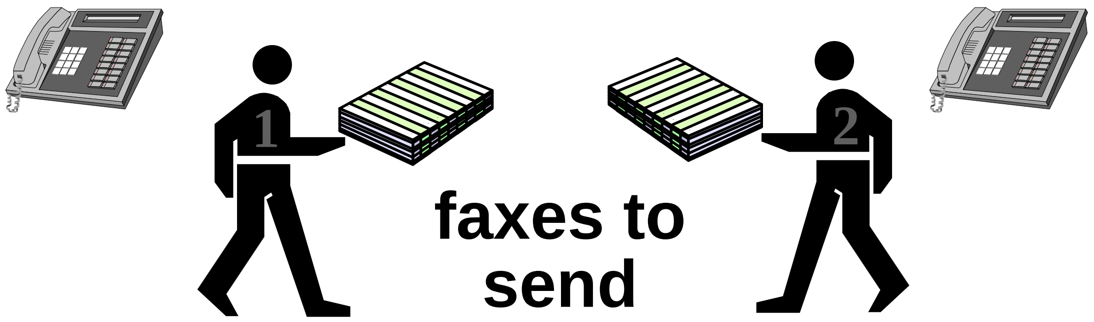

TOD
Common Database
"All female employees with more than three children."
1
"All female employees making over $50,000 a year."
2

Fax Domain



TOD



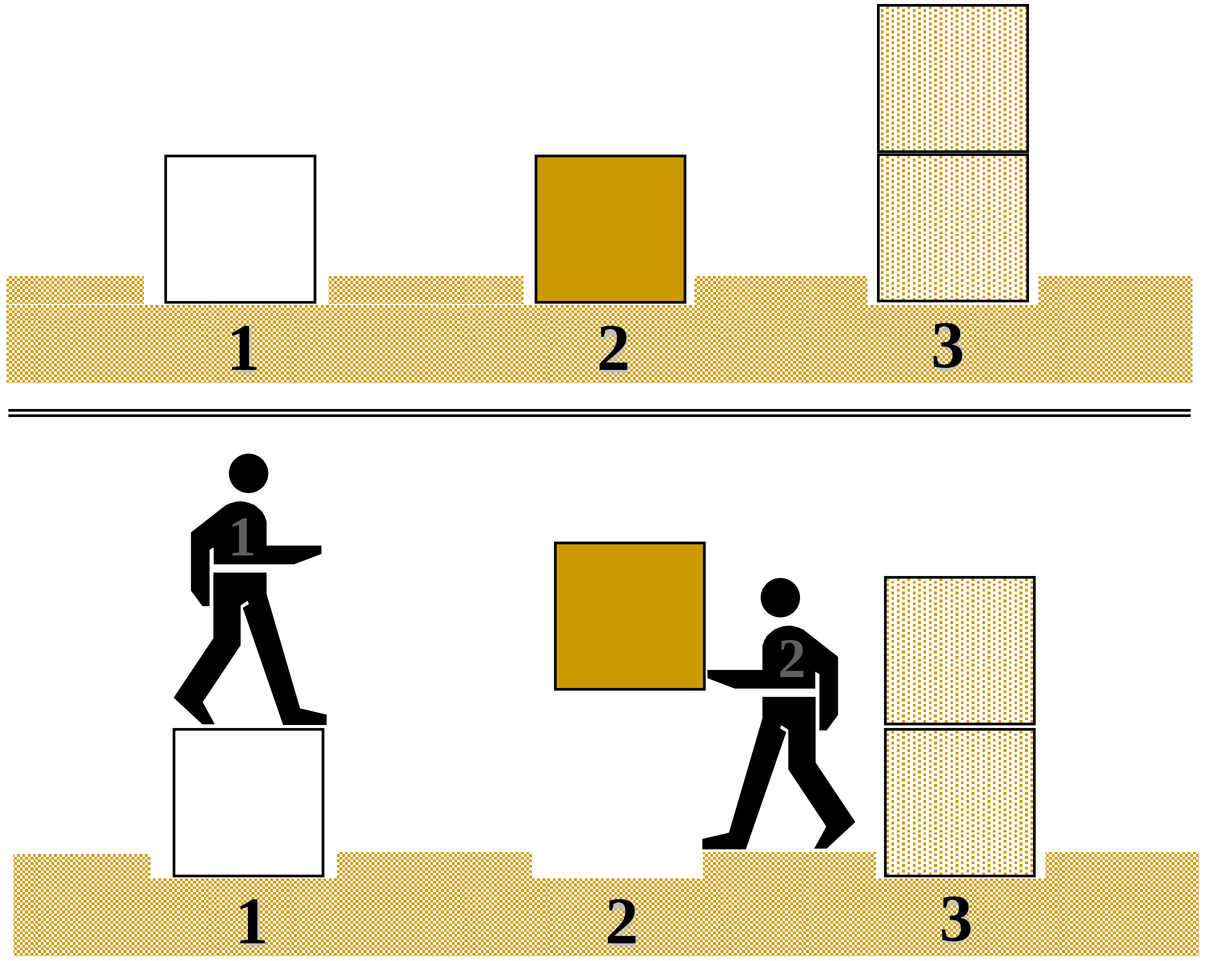


faxes to send

Cost is only to establish connection

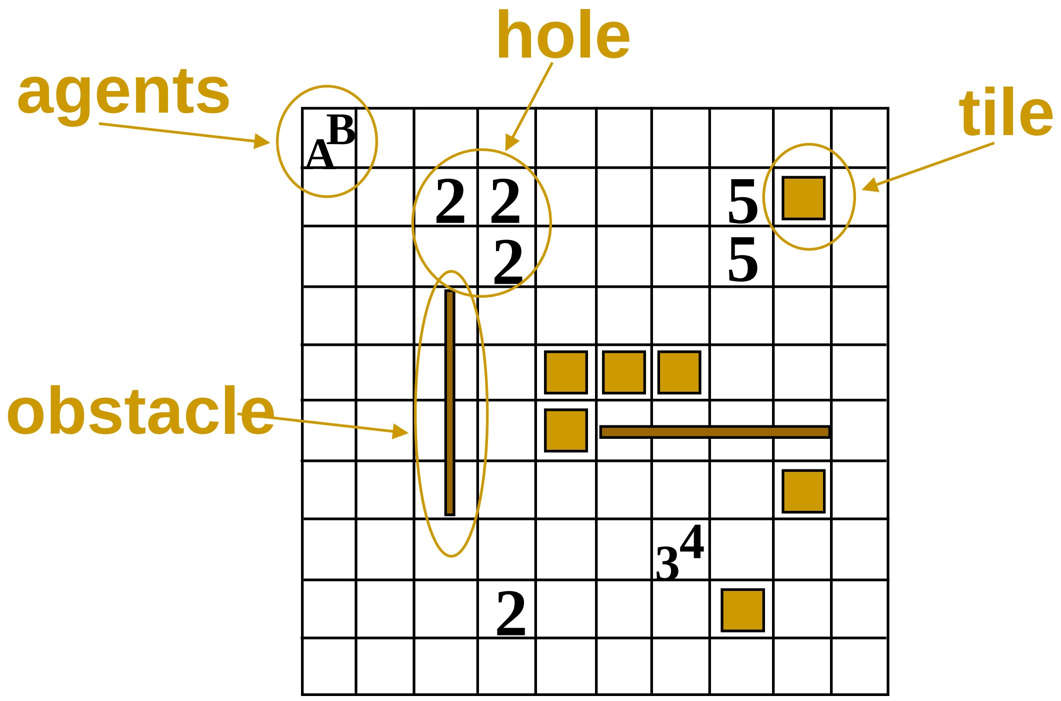
Slotted Blocks World

SOD



The Multi-Agent Tileworld

WOD



TODs Defined

- A TOD is a triple

  $$<T, Ag, c>$$

  where
  - T is the (finite) set of all possible tasks
  - $Ag = \{1,\ldots,n\}$ is the set of participating agents
  - $c = \wp(T) \rightarrow \mathbb{R}^+$ defines the *cost* of executing each subset of tasks
- An *encounter* is a collection of tasks

  $$<T_1,\ldots,T_n>$$

  where $T_i \subseteq T$ for each $i \in Ag$

Building Blocks

- ✓ Domain
  - A precise definition of what a goal is
  - Agent operations
- Negotiation Protocol
  - A definition of a deal
  - A definition of utility
  - A definition of the conflict deal
- Negotiation Strategy
  - In Equilibrium
  - Incentive-compatible

Deals in TODs

- Given encounter $<T_1, T_2>$, a *deal* is an allocation of the tasks $T_1 \cup T_2$ to the agents 1 and 2
- The *cost* to i of deal $\delta = <D_1, D_2>$ is $c(D_i)$, and will be denoted $cost_i(\delta)$
- The *utility* of deal δ to agent i is:

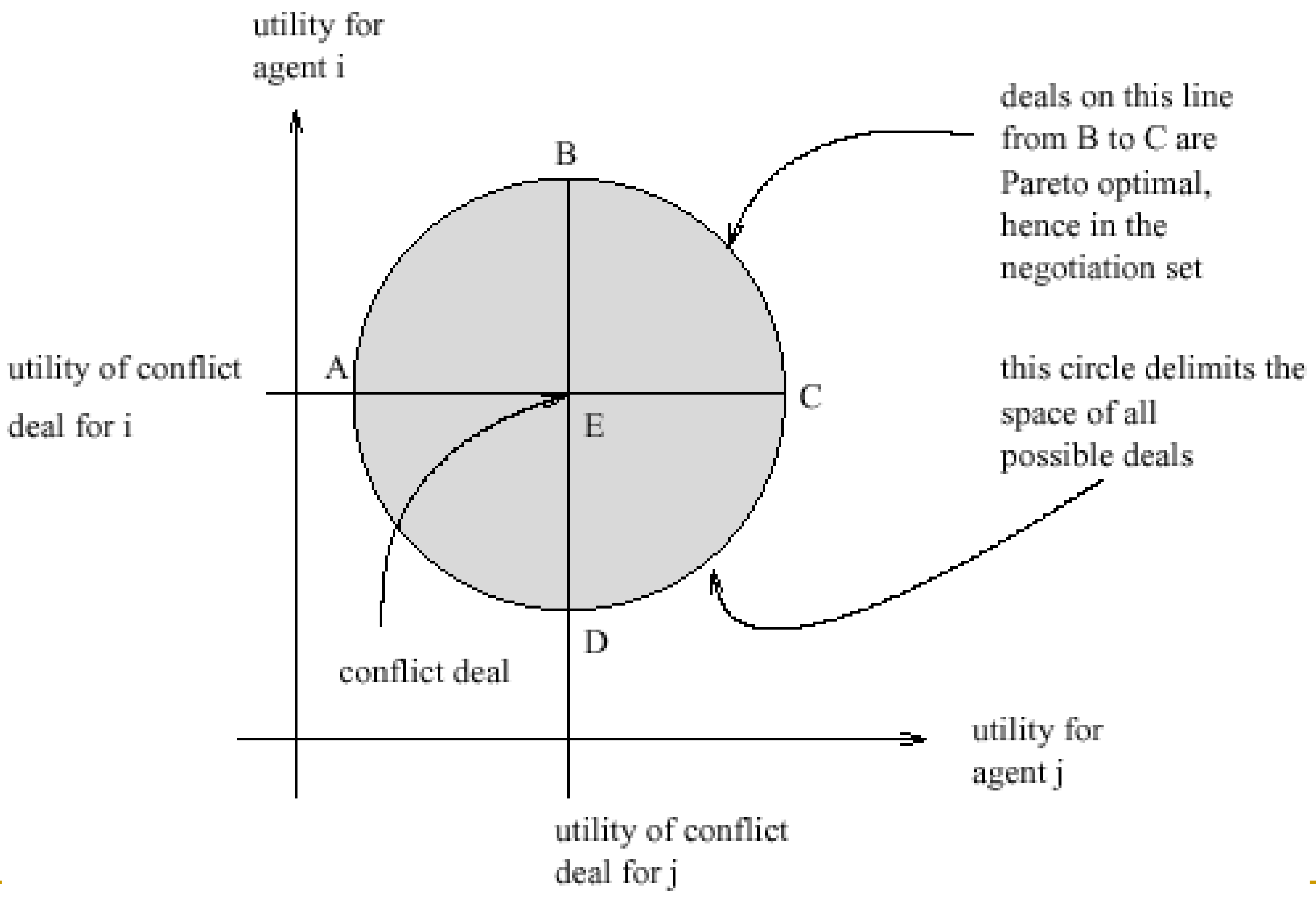
$$utility_i(\delta) = c(T_i) - cost_i(\delta)$$

- The conflict deal, Θ, is the deal $<T_1, T_2>$ consisting of the tasks originally allocated. Note that $utility_i(\Theta) = 0$ for all $i \in Ag$
- Deal δ is *individual rational* if it weakly dominates the conflict deal

The Negotiation Set

- The set of deals over which agents negotiate are those that are:
  - individual rational
  - pareto efficient

The Negotiation Set Illustrated


utility for agent i
deals on this line from B to C are Pareto optimal, hence in the negotiation set
B
A
utility of conflict deal for i
C
E
this circle delimits the space of all possible deals
D
conflict deal
utility for agent j
utility of conflict deal for j

Negotiation Protocols

- Agents use a product-maximizing negotiation protocol (as in Nash bargaining theory)
- It should be a symmetric PMM (product maximizing mechanism)
- Examples: 1-step protocol, monotonic concession protocol…

The Monotonic Concession Protocol

Rules of this protocol are as follows…

- Negotiation proceeds in rounds
- On round 1, agents simultaneously propose a deal from the negotiation set
- Agreement is reached if one agent finds that the deal proposed by the other is at least as good or better than its proposal
- If no agreement is reached, then negotiation proceeds to another round of simultaneous proposals
- In round $u + 1$, no agent is allowed to make a proposal that is less preferred by the other agent than the deal it proposed at time u
- If neither agent makes a concession in some round $u > 0$, then negotiation terminates, with the conflict deal

The Zeuthen Strategy

Three problems:

- What should an agent's first proposal be?
  Its most preferred deal
- On any given round, *who should concede*?
  The agent least willing to risk conflict
- If an agent concedes, then *how much* should it concede?
  Just enough to change the balance of risk

Willingness to Risk Conflict

- Suppose you have conceded a *lot*. Then:
  - Your proposal is now near the conflict deal
  - In case conflict occurs, you are not much worse off
  - You are *more willing* to risk confict
- An agent will be *more willing* to risk conflict if the difference in utility between its current proposal and the conflict deal is *low*

Nash Equilibrium Again…

- The Zeuthen strategy is in Nash equilibrium: under the assumption that one agent is using the strategy the other can do no better than use it himself…
- This is of particular interest to the designer of automated agents. It does away with any need for secrecy on the part of the programmer. An agent's strategy can be publicly known, and no other agent designer can exploit the information by choosing a different strategy. In fact, it is desirable that the strategy be known, to avoid inadvertent conflicts.

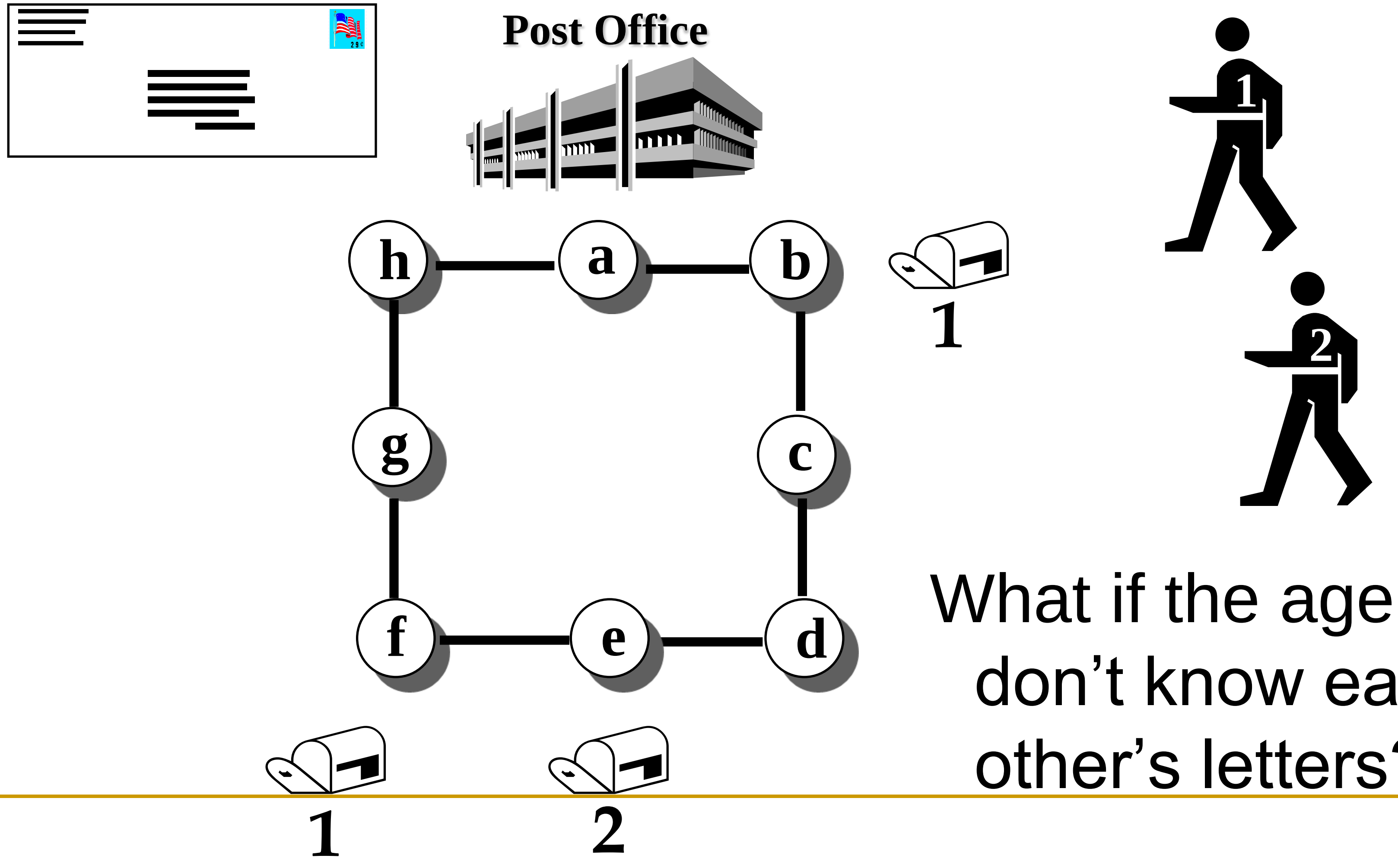
Building Blocks

- ✓ Domain
  - A precise definition of what a goal is
  - Agent operations
- ✓ Negotiation Protocol
  - A definition of a deal
  - A definition of utility
  - A definition of the conflict deal
- Negotiation Strategy
  - In Equilibrium
  - Incentive-compatible

Deception in TODs

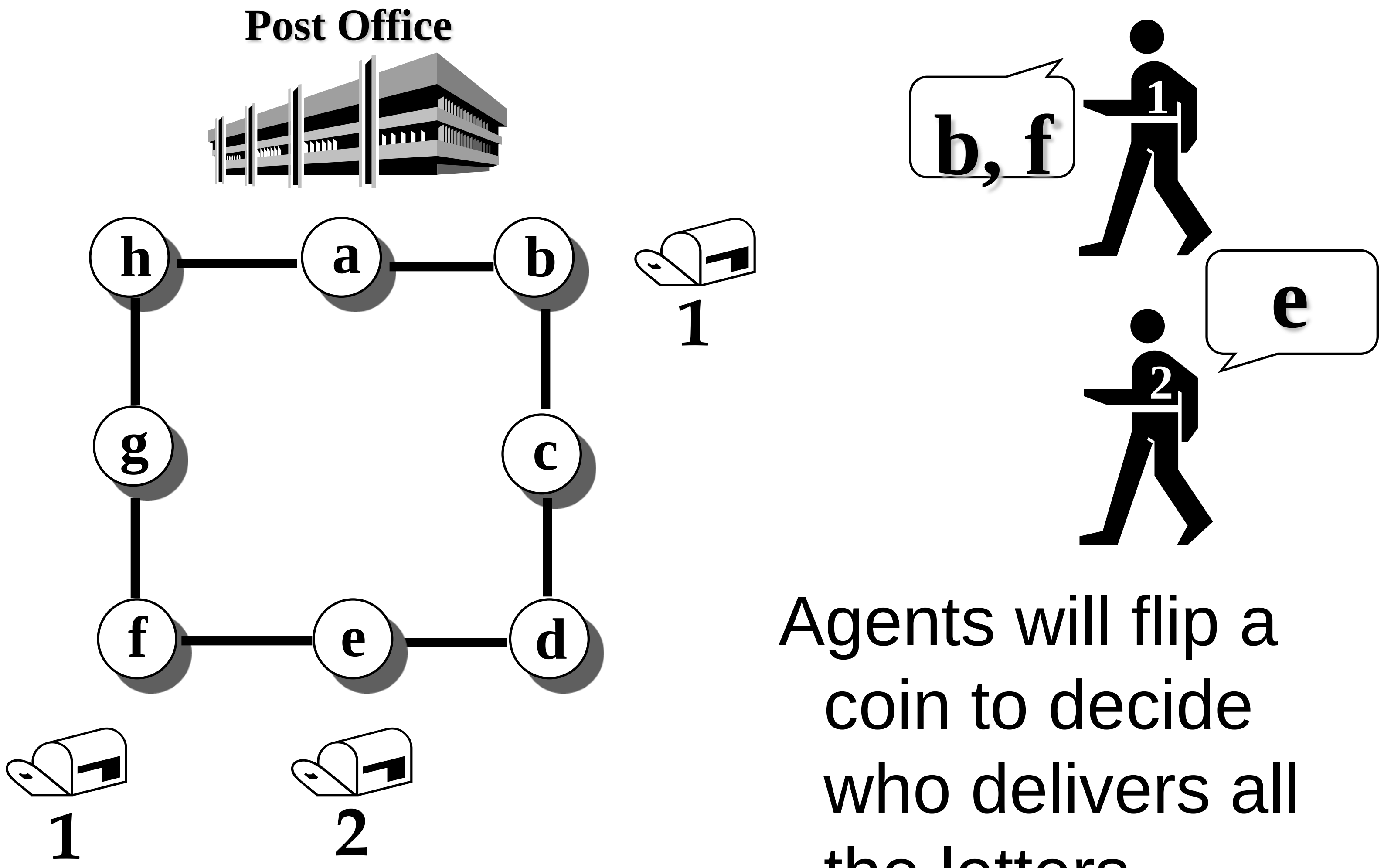
- Deception can benefit agents in two ways:
  - *Phantom and Decoy tasks*
    Pretending that you have been allocated tasks you have not
  - *Hidden tasks*
    Pretending *not* to have been allocated tasks that you have been

Negotiation with Incomplete Information



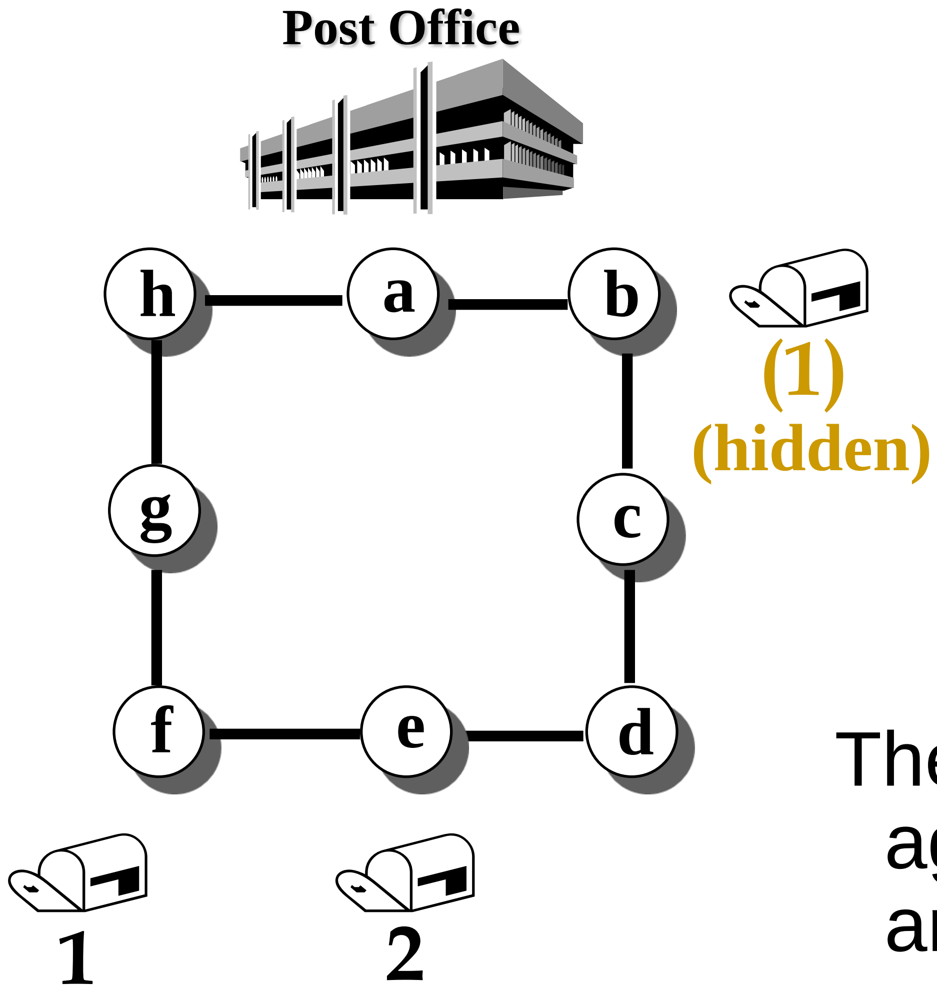
What if the agents don’t know each other’s letters?

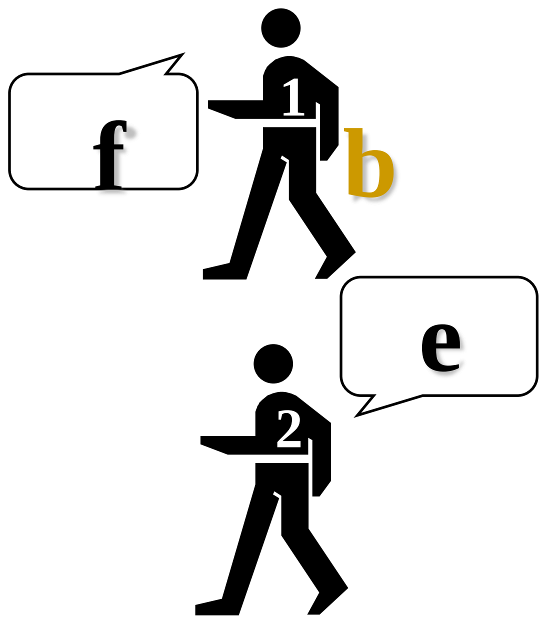
–1 Phase Game: Broadcast Tasks



Agents will flip a coin to decide who delivers all the letters

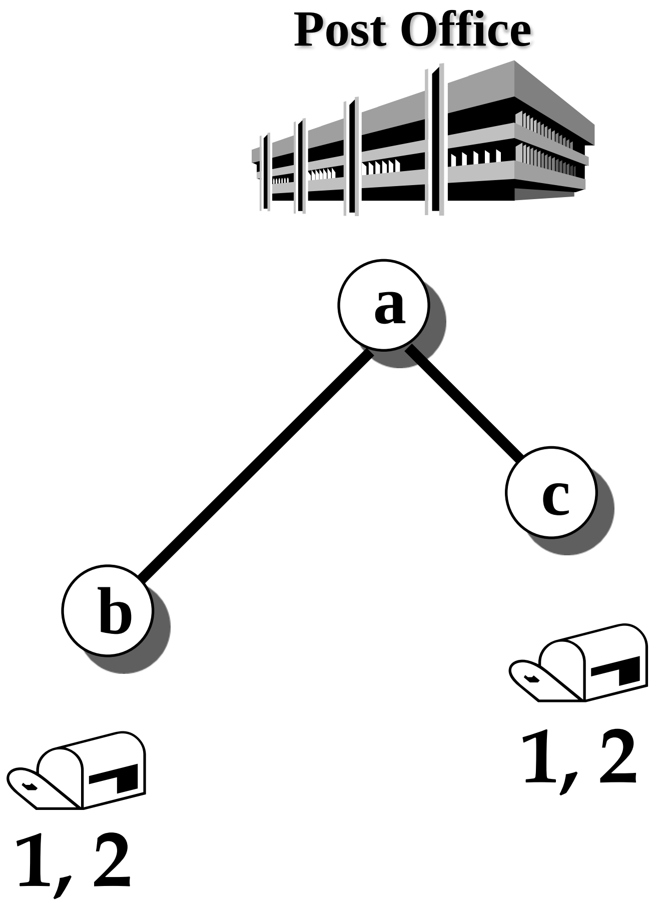
Hiding Letters

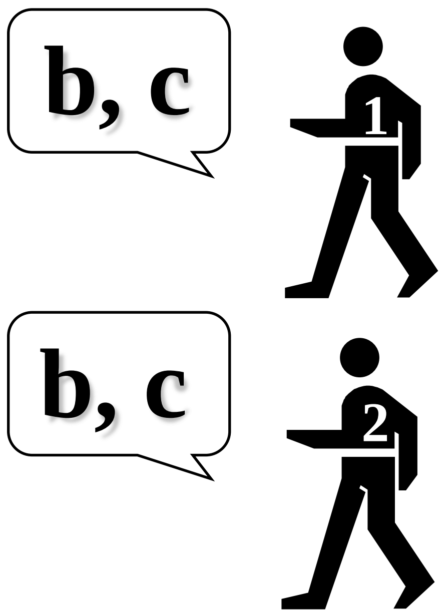




They then agree that agent 2 delivers to f and e

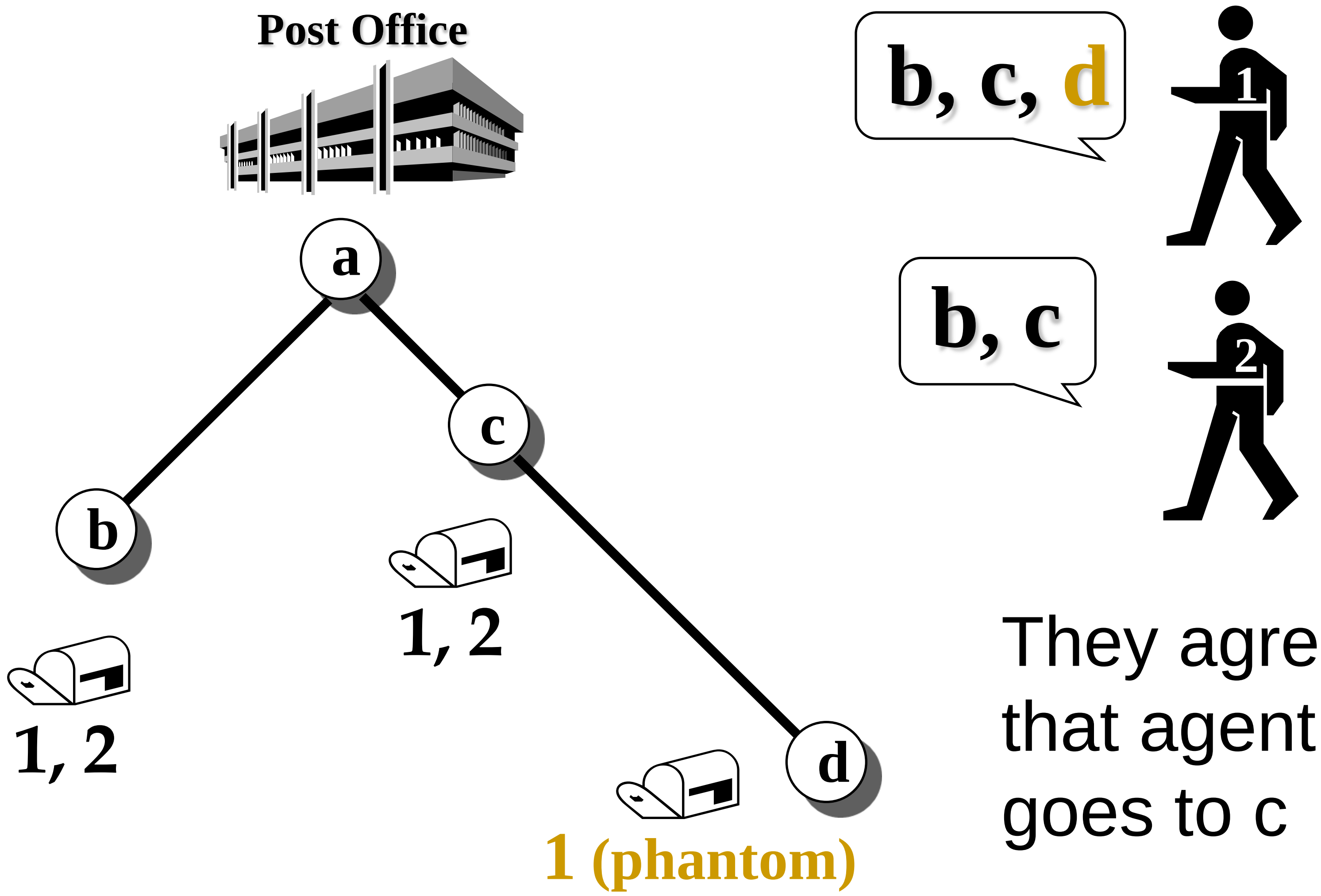
Another Possibility for Deception





They will agree to flip a coin to decide who goes to b and who goes to c

Phantom Letter



They agree that agent 1 goes to c

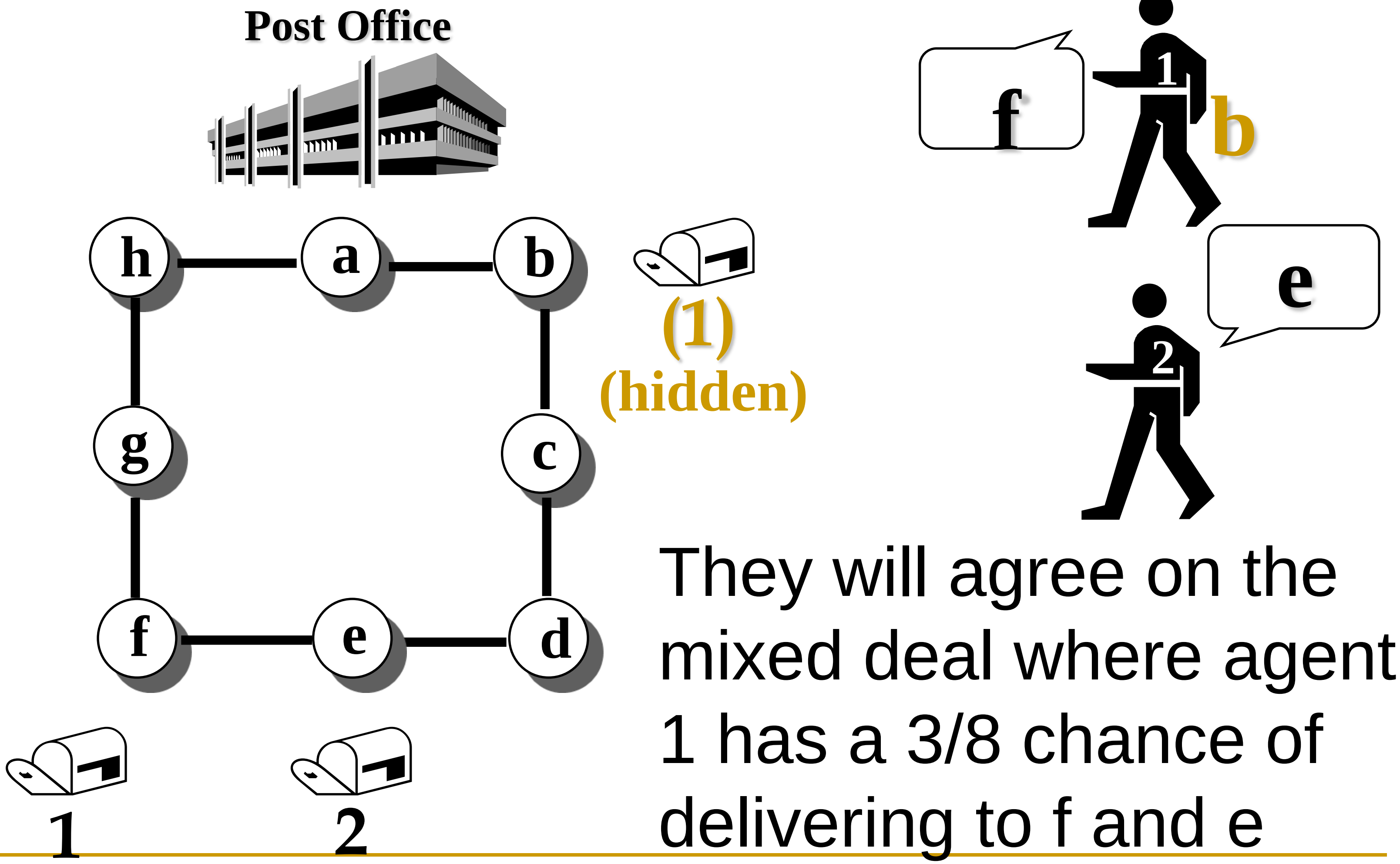
Negotiation over Mixed Deals

Mixed deal $<D_1, D_2> : p$

The agents will perform $<D_1, D_2>$ with probability p, and the symmetric deal $<D_2, D_1>$ with probability $1 – p$

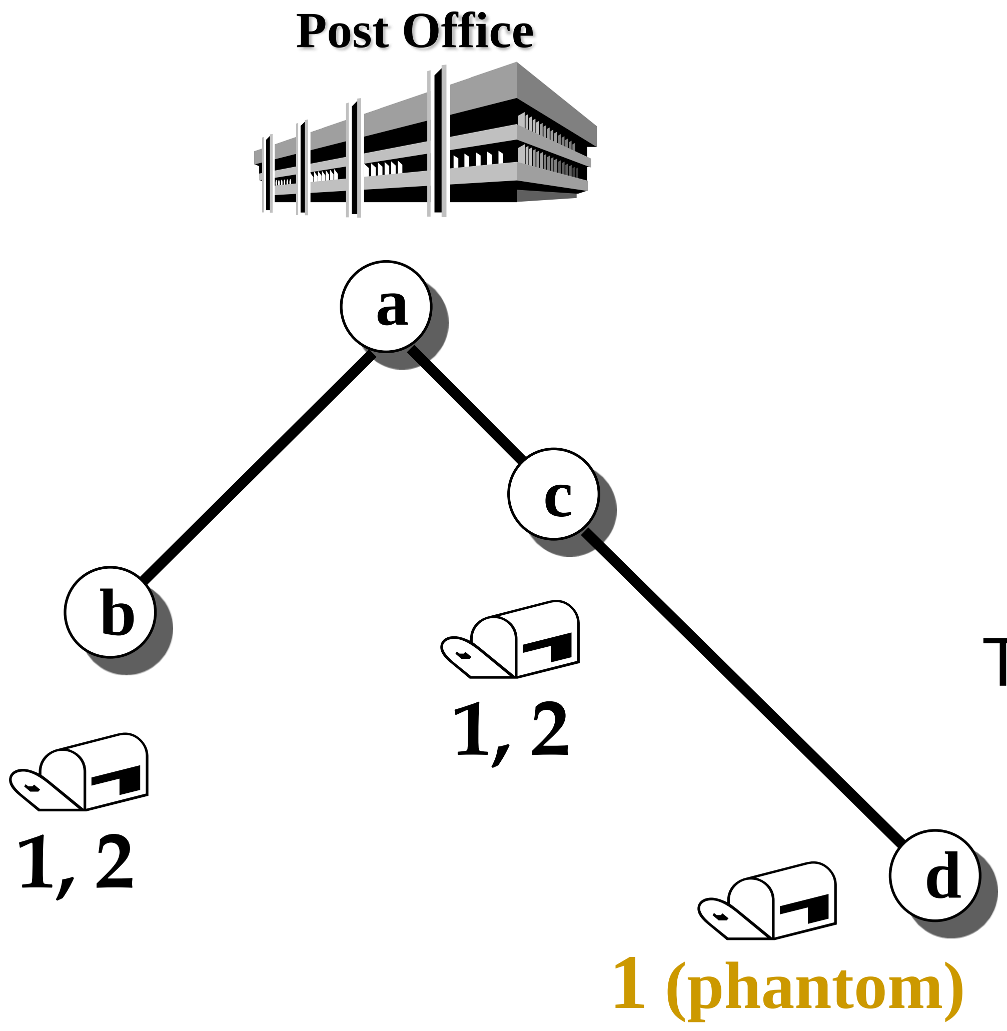
Theorem: With mixed deals, agents can always agree on the “all-or-nothing” deal – where D_1 is $T_1 \cup T_2$ and D_2 is the empty set

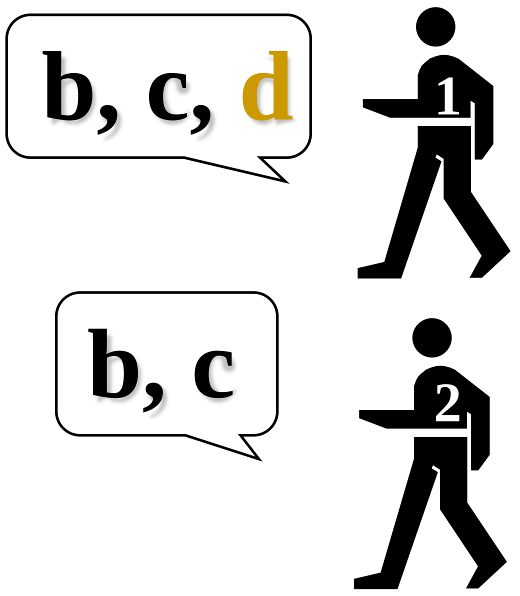
Hiding Letters with Mixed All-or-Nothing Deals



They will agree on the mixed deal where agent 1 has a 3/8 chance of delivering to f and e

Phantom Letters with Mixed Deals





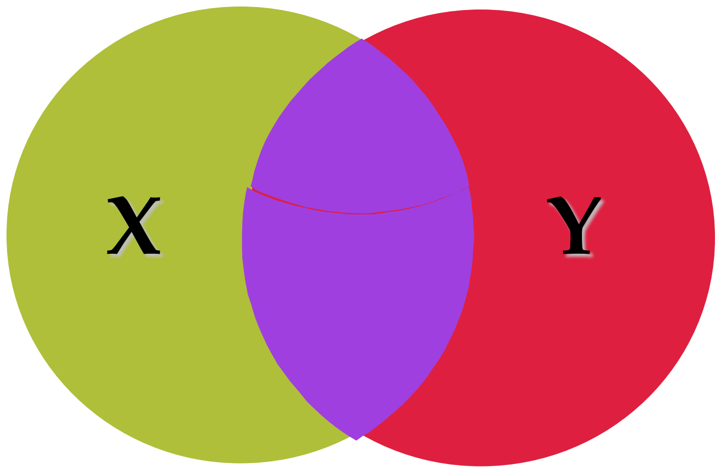
They will agree on the mixed deal where A has 3/4 chance of delivering all letters, lowering his expected utility

Sub-Additive TODs

TOD < T, Ag, c > is *sub-additive* if for all finite sets of tasks X, Y in T we have:

$$c(X \cup Y) \leq c(X) + c(Y)$$

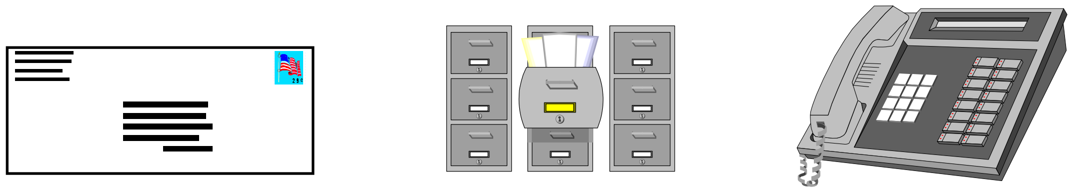
Sub-Additivity

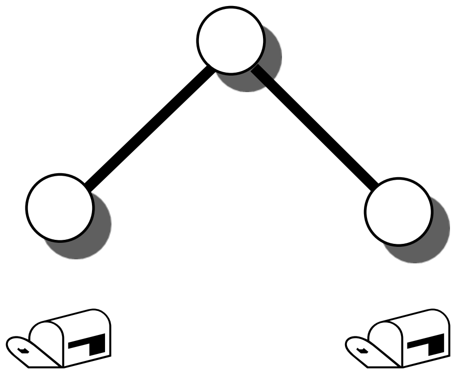


$$c(\boldsymbol{X} \cup \boldsymbol{Y}) \leq c(\boldsymbol{X}) + c(\boldsymbol{Y})$$

Sub-Additive TODs

The Postmen Domain, Database Domain, and Fax Domain are sub-additive.





The "Delivery Domain" (where postmen don't have to return to the Post Office) is not sub-additive

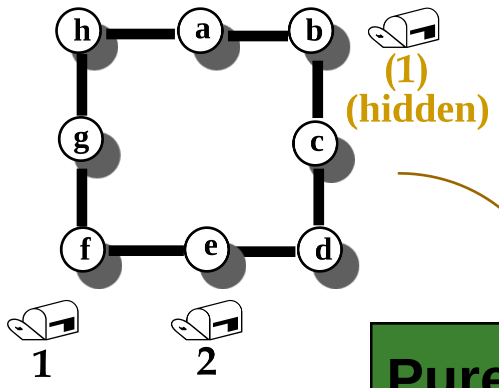
Incentive Compatible Mechanisms

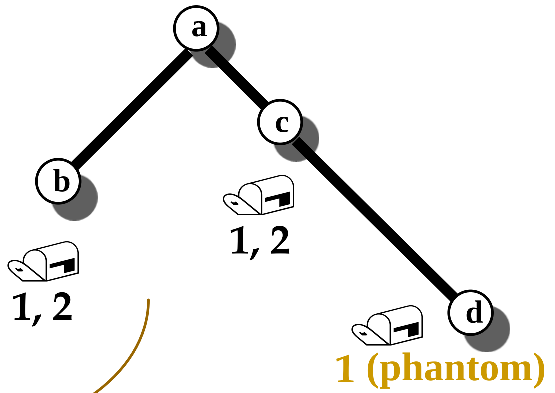
Sub-Additive

	Hidden	Phantom
Pure	L	L
A/N	T	T/P ↑
Mix	L	T/P

- L means “there exists a beneficial lie in some encounter”
- T means “truth telling is dominant, there never exists a beneficial lie, for all encounters”
- T/P means “truth telling is dominant, if a discovered lie carries a sufficient penalty”
- A/N signifies all-or-nothing mixed deals

Incentive Compatible Mechanisms





Sub-Additive

	Hidden	Phantom
Pure	L	L
A/N	T	T/P
Mix	L	T/P

***Theorem*: For all encounters in all sub-additive TODs, when using a PMM over all-or-nothing deals, no agent has an incentive to hide a task.**

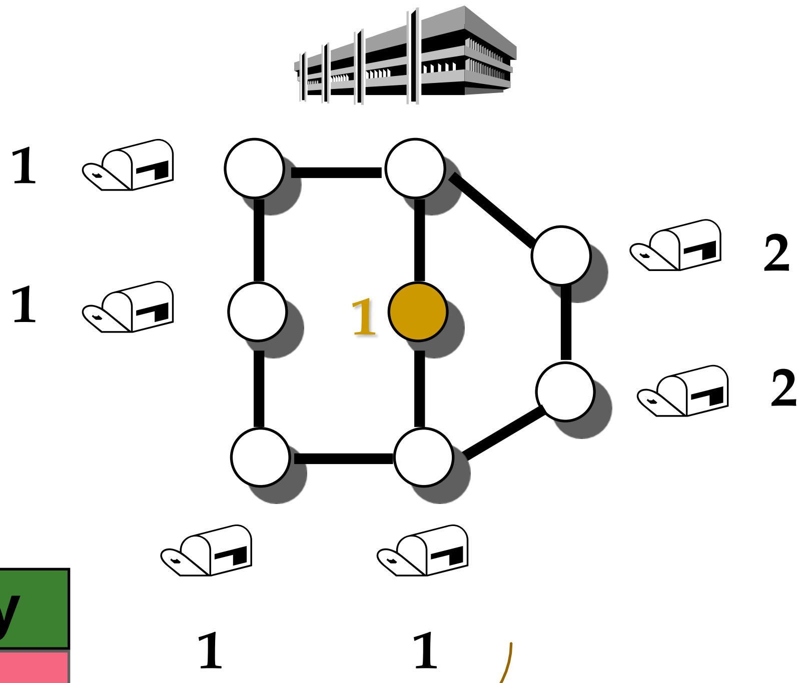
Incentive Compatible Mechanisms

	Hidden	Phantom
Pure	L	L
A/N	T	T/P ↑
Mix	L	T/P

- Explanation of the up-arrow:
  If it is never beneficial in a *mixed* deal encounter to use a phantom lie (with penalties), then it is certainly never beneficial to do so in an all-or-nothing mixed deal encounter (which is just a subset of the mixed deal encounters)

Decoy Tasks

Decoy tasks, however, can be beneficial even with all-or-nothing deals



Sub-Additive

	Hidden	Phantom	Decoy
Pure	L	L →	L
A/N	T	T/P ↑	L ↓
Mix	L	T/P	L

Decoy lies are simply phantom lies where the agent is able to manufacture the task (if necessary) to avoid discovery of the lie by the other agent.

Decoy Tasks

Sub-Additive

	Hidden	Phantom	Decoy
Pure	L	L →	L
A/N	T	T/P ↑	L ↓
Mix	L	T/P	L

- Explanation of the down arrow:
  If there exists a beneficial decoy lie in some all-or-nothing mixed deal encounter, then there certainly exists a beneficial decoy lie in some general mixed deal encounter (since all-or-nothing mixed deals are just a subset of general mixed deals)

Decoy Tasks

Sub-Additive

	Hidden	Phantom	Decoy
Pure	L	L ⟶	L
A/N	T	T/P ↑	L ↓
Mix	L	T/P	L

- Explanation of the horizontal arrow:
  If there exists a beneficial phantom lie in some pure deal encounter, then there certainly exists a beneficial decoy lie in some pure deal encounter (since decoy lies are simply phantom lies where the agent is able to manufacture the task if necessary)

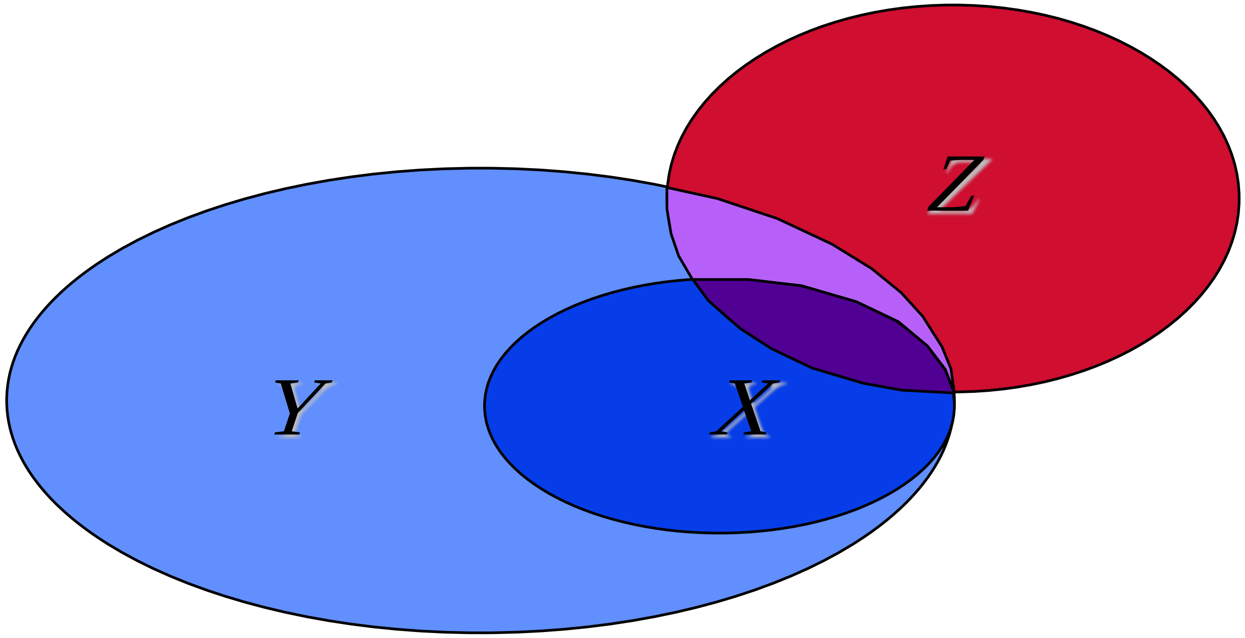
Concave TODs

TOD $< T, Ag, c >$ is *concave* if for all finite sets of tasks Y and Z in T, and $X \subseteq Y$, we have:

$$c(Y \cup Z) - c(Y) \leq c(X \cup Z) - c(X)$$

Concavity implies sub-additivity

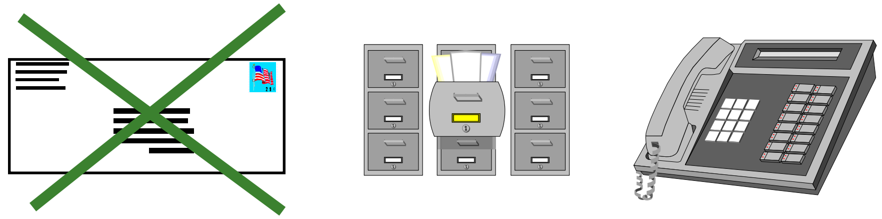
Concavity

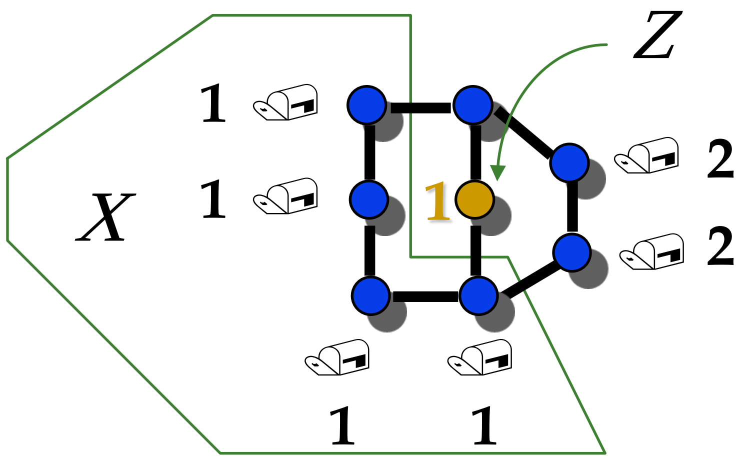


The cost Z adds to X is more than the cost it adds to Y.
(Z - X is a superset of Z - Y)

Concave TODs

The Database Domain and Fax Domain are concave (not the Postmen Domain, unless restricted to trees).





This example was not concave; Z adds 0 to X, but adds 2 to its superset Y (all blue nodes)

Three-Dimensional Incentive Compatible Mechanism Table

***Theorem*: For all encounters in all concave TODs, when using a PMM over all-or-nothing deals, no agent has any incentive to lie.**

Concave

	Hidden	Phantom	Decoy
Pure	L	L	L
A/N	T	T	T
Mix	L	T	T

Sub-Additive

	Hidden	Phantom	Decoy
Pure	L	L	L
A/N	T	T/P	L
Mix	L	T/P	L

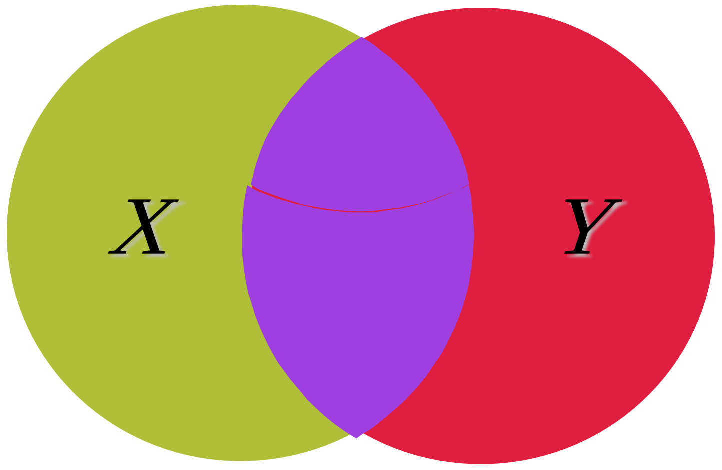
Modular TODs

TOD < T, Ag, c > is *modular* if for all finite sets of tasks X, Y in T we have:

$$c(X \cup Y) = c(X) + c(Y) - c(X \cap Y)$$

Modularity implies concavity

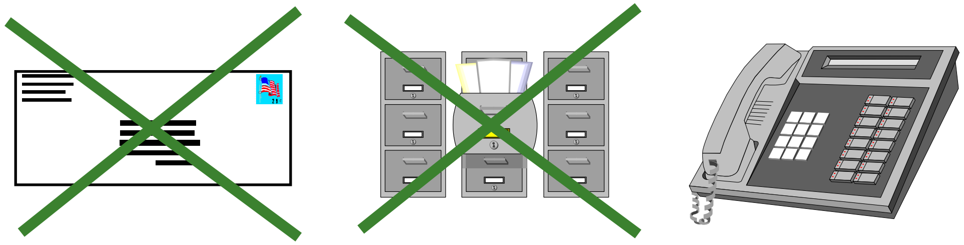
Modularity



$$c(X \cup Y) = c(X) + c(Y) - c(X \cap Y)$$

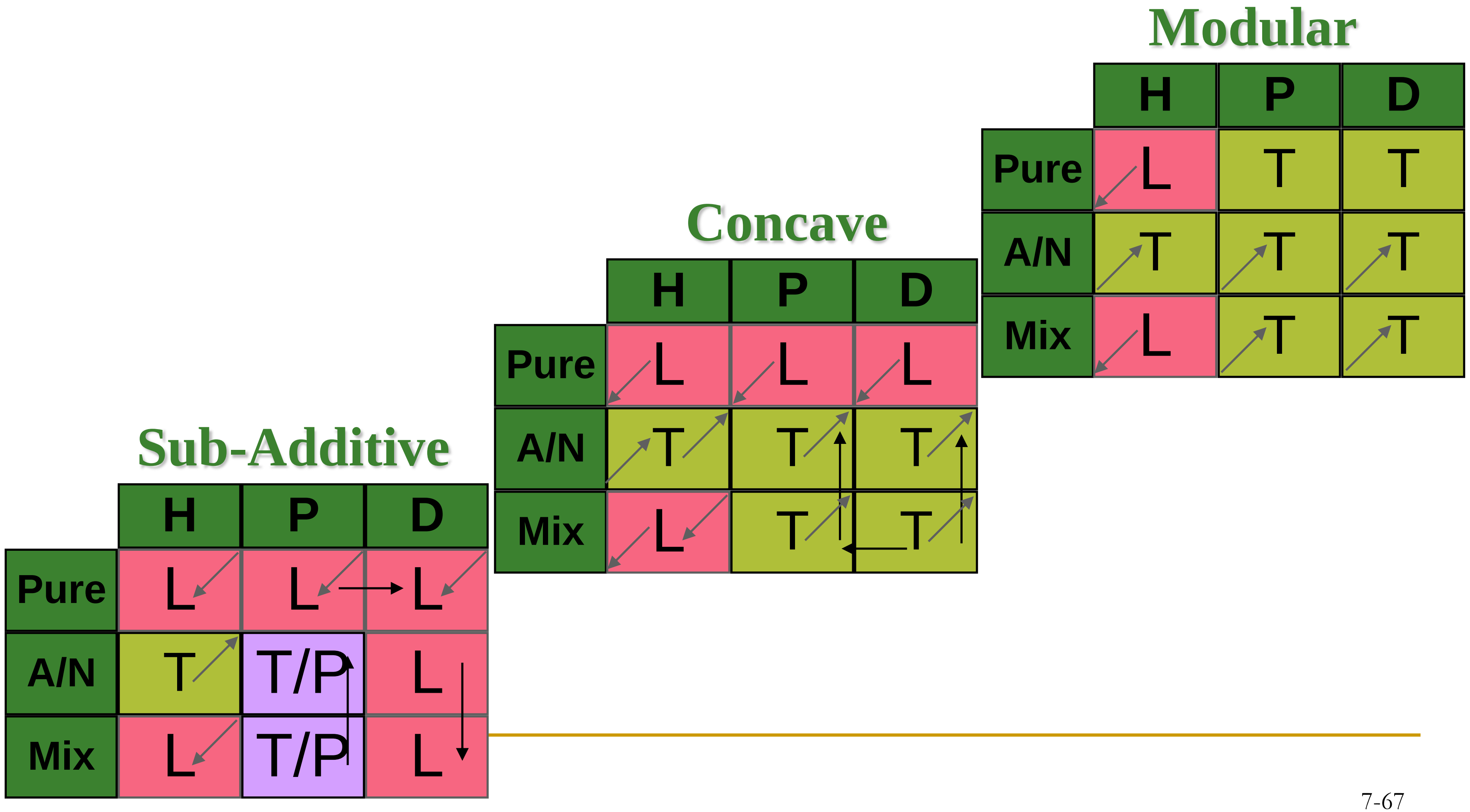
Modular TODs

The Fax Domain is modular (not the Database Domain nor the Postmen Domain, unless restricted to a star topology).



Even in modular TODs, hiding tasks can be beneficial in general mixed deals

Three-Dimensional Incentive Compatible Mechanism Table



Sub-Additive

	H	P	D
Pure	L	L	L
A/N	T	T/P	L
Mix	L	T/P	L

Concave

	H	P	D
Pure	L	L	L
A/N	T	T	T
Mix	L	T	T

Modular

	H	P	D
Pure	L	T	T
A/N	T	T	T
Mix	L	T	T

Related Work

- Similar analysis made of State Oriented Domains, where situation is more complicated
- Coalitions (more than two agents, Kraus, Shechory)
- Mechanism design (Sandholm, Nisan, Tennenholtz, Ephrati, Kraus)
- Other models of negotiation (Kraus, Sycara, Durfee, Lesser, Gasser, Gmytrasiewicz)
- Consensus mechanisms, voting techniques, economic models (Wellman, Ephrati)

Conclusions

- By appropriately adjusting the *rules of encounter* by which agents must interact, we can influence the private strategies that designers build into their machines

Rules of Encounter

- The interaction mechanism should ensure the *efficiency* of multi-agent systems

Efficiency

Conclusions

- To maintain efficiency over time of dynamic multi-agent systems, the rules must also be *stable*
- The use of formal tools enables the design of efficient and stable mechanisms, and the precise characterization of their properties

Stability

Formal Tools

Argumentation

- Argumentation is the process of attempting to convince others of something
- Gilbert (1994) identified 4 modes of argument:

1. *Logical mode*
   “If you accept that *A* and that *A* implies *B*, then you must accept that *B*”
2. *Emotional mode*
   “How would you feel if it happened to you?”
3. *Visceral mode*
   “Cretin!”
4. *Kisceral mode*
   “This is against Christian teaching!”

Logic-based Argumentation

Basic form of logical arguments is as follows:

Database ✿ (*Sentence, Grounds*)

where:

- *Database* is a (possibly inconsistent) set of logical formulae
- *Sentence* is a logical formula known as the *conclusion*
- *Grounds* is a set of logical formulae such that:
  1. *Grounds* ♐ *Database*; and
  2. *Sentence* can be proved from *Grounds*

Attack and Defeat

- Let (ϕ_1, Γ_1) and (ϕ_2, Γ_2) be arguments from some database Δ…
  Then (ϕ_2, Γ_2) can be defeated (attacked) in one of two ways:
- (ϕ_1, Γ_1) *rebuts* (ϕ_2, Γ_2) if ϕ_1 📪 $\neg\phi_2$
- (ϕ_1, Γ_1) *undercuts* (ϕ_2, Γ_2) if ϕ_1 📪 $\neg\psi_2$ for some ψ 🗀 Γ_2

A rebuttal or undercut is known as an *attack*

Abstract Argumentation

- Concerned with the overall structure of the argument (rather than internals of arguments)
- Write $x \rightarrow y$
  - "argument x attacks argument y"
  - "x is a counterexample of y"
  - "x is an attacker of y"

where we are not actually concerned as to what x, y are

- An *abstract argument system* is a collection or arguments together with a relation "$\rightarrow$" saying what attacks what
- An argument is *out* if it has an undefeated attacker, and *in* if all its attackers are defeated

An Example Abstract Argument System

$$\begin{tikzcd}
 & & c \arrow[ddll, bend right] \arrow[dl] & & & & & & m \arrow[llllll, bend right] \arrow[dl] & \\
 & d \arrow[dl] \arrow[dd] \arrow[dddrr, bend left] & & g \arrow[ll] \arrow[ddrrrrr, bend left] & & & & k \arrow[rr, bend right] & & l \arrow[ul, bend right] \\
a & & & & & j \arrow[dl, bend left] \arrow[ddr, bend left] & & & & \\
 & b & & & i \arrow[ur, bend left] \arrow[dl] \arrow[drr, bend right] & & & & p \arrow[uuullllll] \arrow[uur, bend right] \arrow[ddr, bend left] & \\
 & & & e \arrow[ull] & & & n \arrow[dll] \arrow[urr, bend right] & & & \\
 & & & & f & & & & & q \\
 & & & h \arrow[uu] \arrow[uuuulll, bend left] \arrow[uuurrrrr, bend right] & & & & & &
\end{tikzcd}$$