Intelligent Agents
Negotiation and Rules of Encounter

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

Strategies of Agents and Game Theory

- Given a set of agents, their preferences, and an agreed protocol, the final ingredient is the agent's strategy
- The strategy may specify what offer to make next or what information to reveal (truthfully or not)
- A rational agent's strategy must aim to achieve the best possible outcome for him/her
- Game-theory is used to analyze agents' strategic behavior

Discussed optimal strategies for auctions last time

Negotiation, Bargaining Problems

- Auctions are *only* concerned with the allocation of goods
- Negotiation is the process of reaching agreements on matters of common interest (bargaining problem)
  - E.g., who does what?

Negotiation, Mechanisms, Strategies, Deals

- Negotiations governed by mechanism (or protocol)
  - Rules of encounter between the agents
  - Public rules by which the agents will come to agreements
  - Strategies that agents should use
  - Deals that can be made
  - Sequence of offers and counter-offers that can be made
- Negotiations can involve
  - Exchange of information (cf. value of information)
  - Relaxation of initial goals
  - Mutual concession (e.g., concerning division of labor)

Negotiation in Applications

Fosuc on TODs

- Task-oriented domains (TOD)
  - Each agent is associated with a set of tasks (e.g., web mining tasks)
  - Goal: redistribute tasks such that overall costs of completing the tasks is reduced/minimized
- State-oriented domains (SOD ⊇ TOD)
  - Each agent has a set of goal states it would like to achieve
  - Use negotiation to achieve a common goal (actions can have positive or negative side effects)
- Worth-oriented domains (WOD ⊇ SOD)
  - Agents assign worth to state (agent-local utility)
  - Goal: maximize mutual worth / compromise on goals

How many agents?

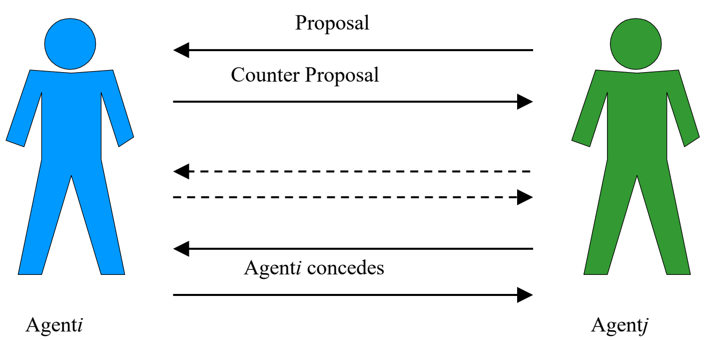
- One to one
- One to many
  (auction as an example of one seller and many buyers)
- Many to many (could be divided into buyers and sellers, or all could be identical in role – like officemate)
  - n(n-1)/2 number of pairs

Criteria of a Negotiation Protocols

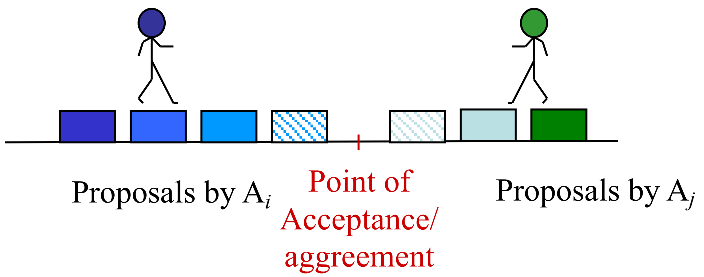
- Efficiency
  - Do not waste utility (compute optimal solution)
- Stability
  - No agent has an incentive to deviate from dominant strategy
- Simplicity
  - Low computational demands on agents (e.g., no counter-speculation required → "dominant strategy" exists)
- Distribution
  - No central decision maker
- Symmetry (possibly)
  - May not want agents to play different roles

Negotiation Process

- Negotiation usually proceeds in a series of rounds, with every agent making a proposal at every round.
- Communication during negotiation:



- Another way of looking at the negotiation process: Who "moves" the farthest



Types of Deals

- Conflict: Keep the same tasks as had originally
- Pure: Divide up tasks
- Mixed: Divide up the tasks, but decide probabilistically who should do what
- All or Nothing (A/N): Mixed deal, with added requirement that we only have all or nothing deals (one of the tasks sets is empty)

Task-Oriented Domain (TOD)

- A task-oriented domain 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 cost of executing subsets of tasks

- Constraints on the cost function c:
  - If $T \subseteq T'$, then $c(T) \leq c(T')$ (monotonicity).
  - $c(\emptyset) = 0$

The Case of Two Agents

- Let (T_1, T_2) be the original tasks of two agents and let $\delta = (D_1, D_2)$ be a new task allocation (a *deal*), i.e.,

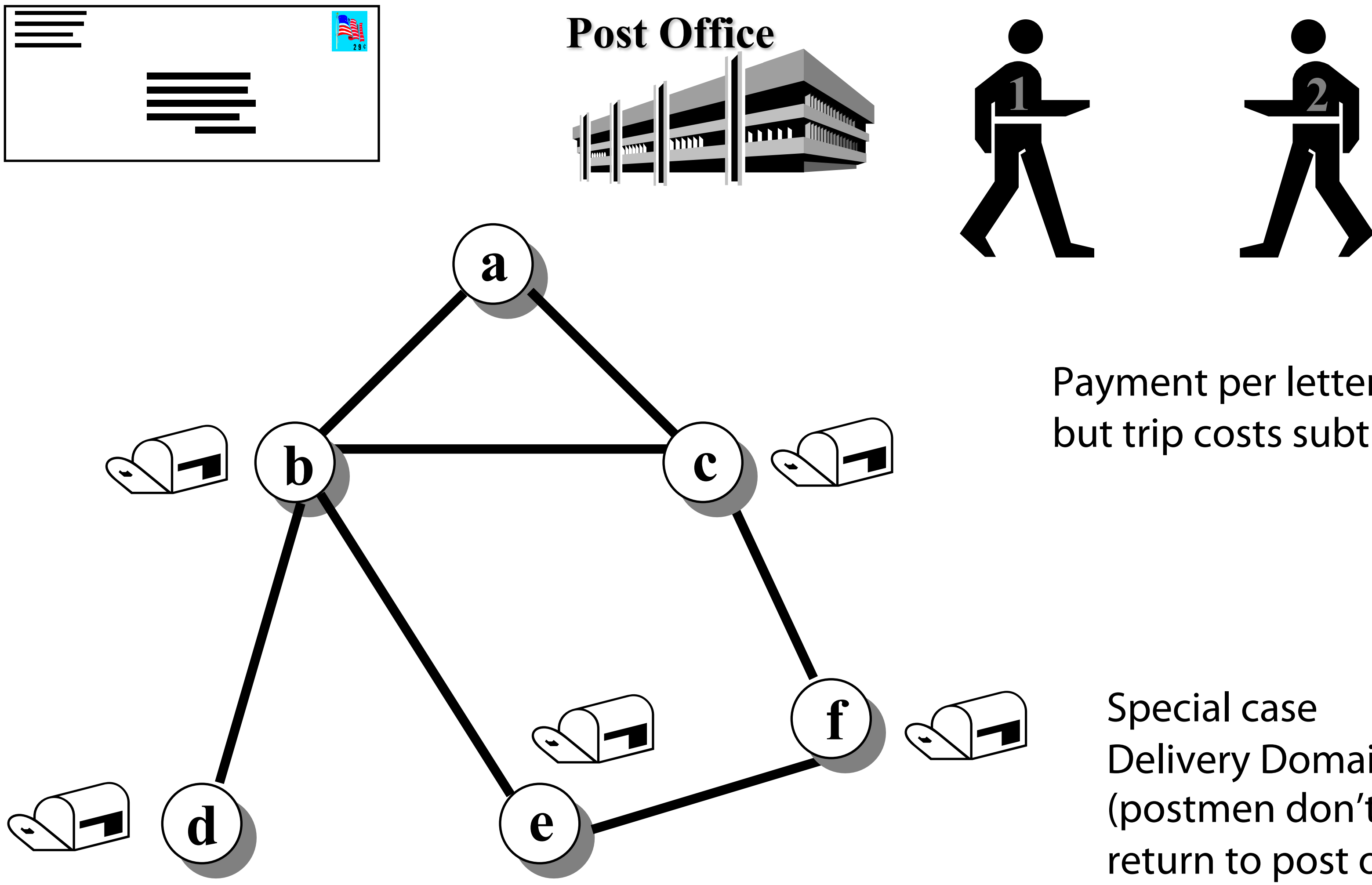
$$T_1 \cup T_2 = D_1 \cup D_2$$

- An agent i's utility of a deal δ is defined as follows:

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

- δ_1 dominates δ_2 when one agent is better off and none is worse off
- Pareto optimal deals: non-dominated deals

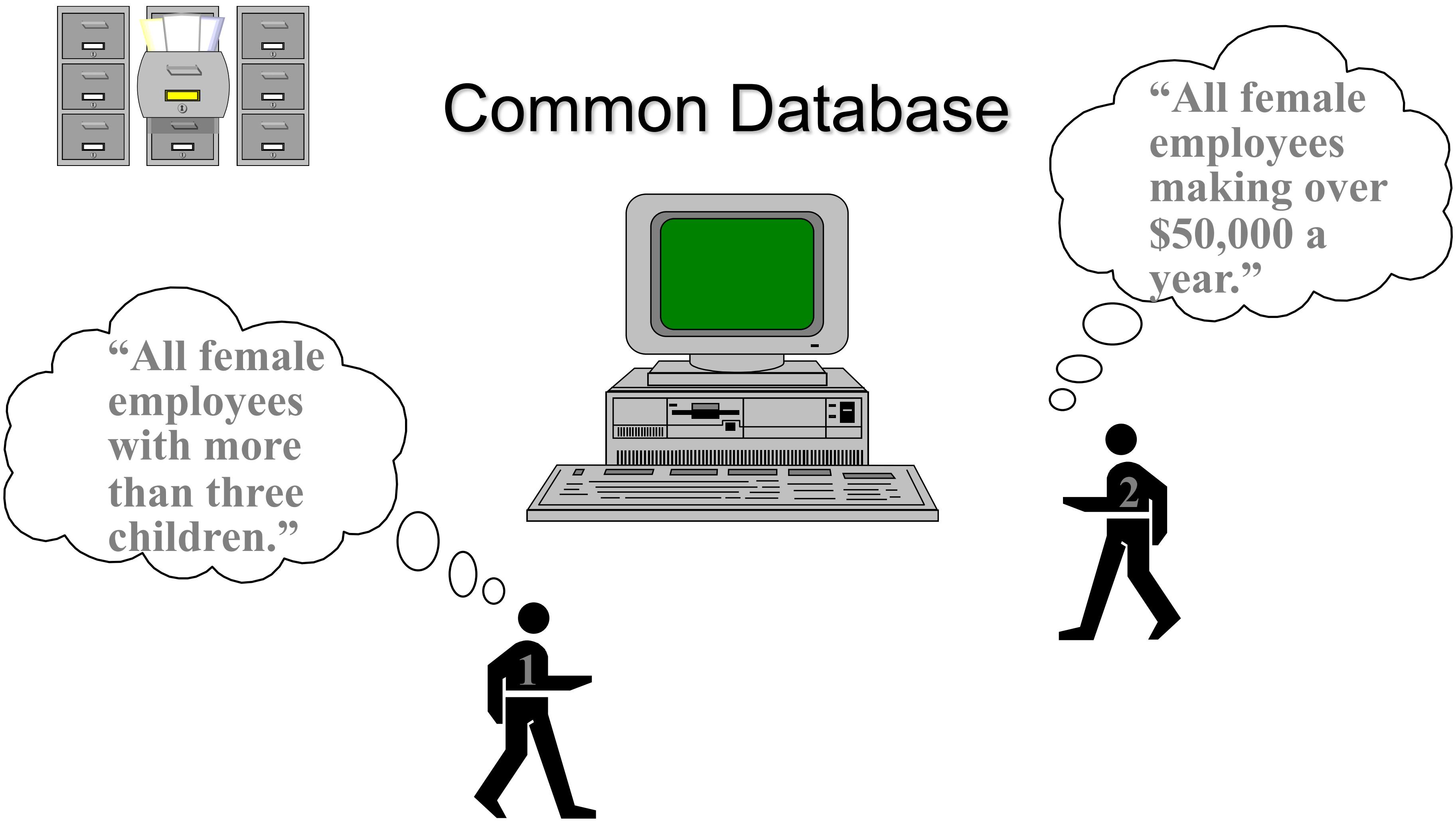
TOD: Postmen Domain



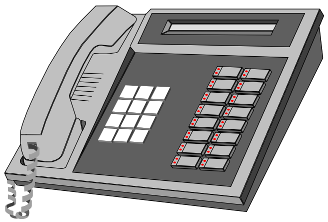
Payment per letter
but trip costs subtracted

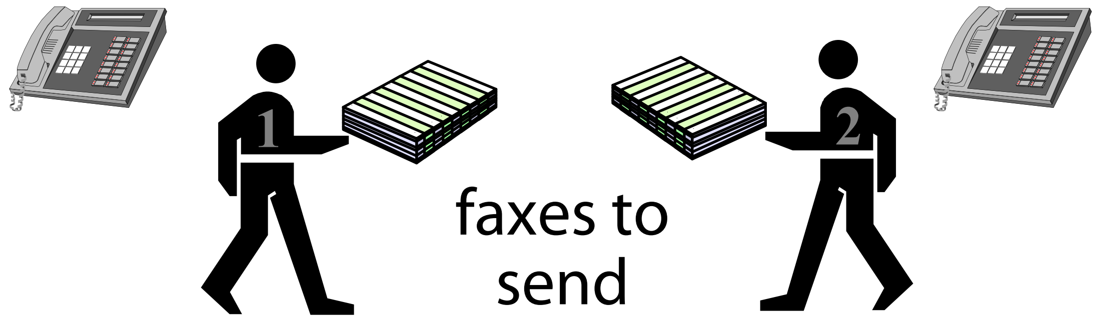
Special case
Delivery Domain
(postmen don't have to return to post office)

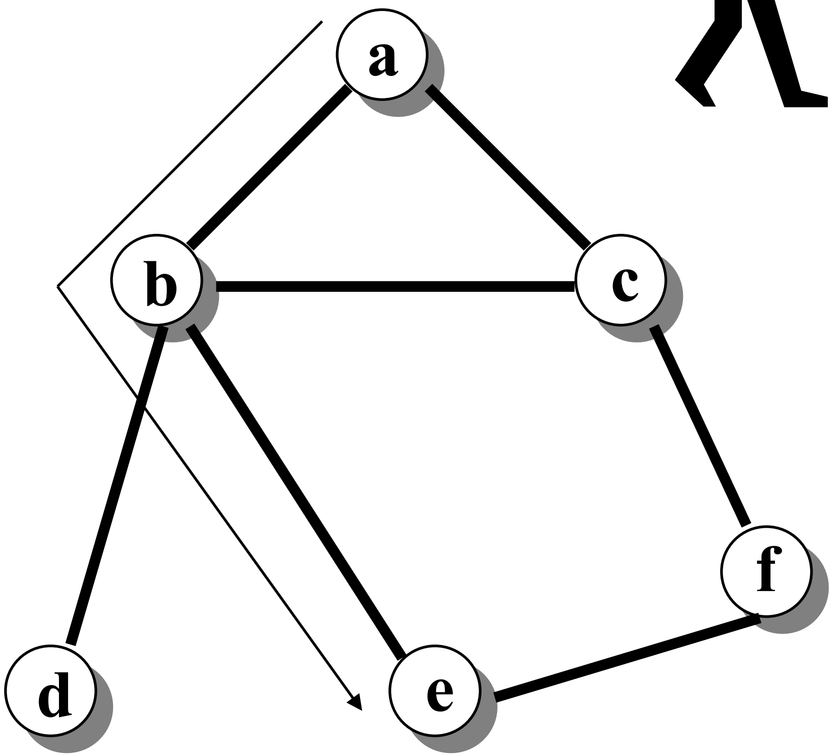
TOD: Database/Web Mining Domain



TOD: Fax Domain

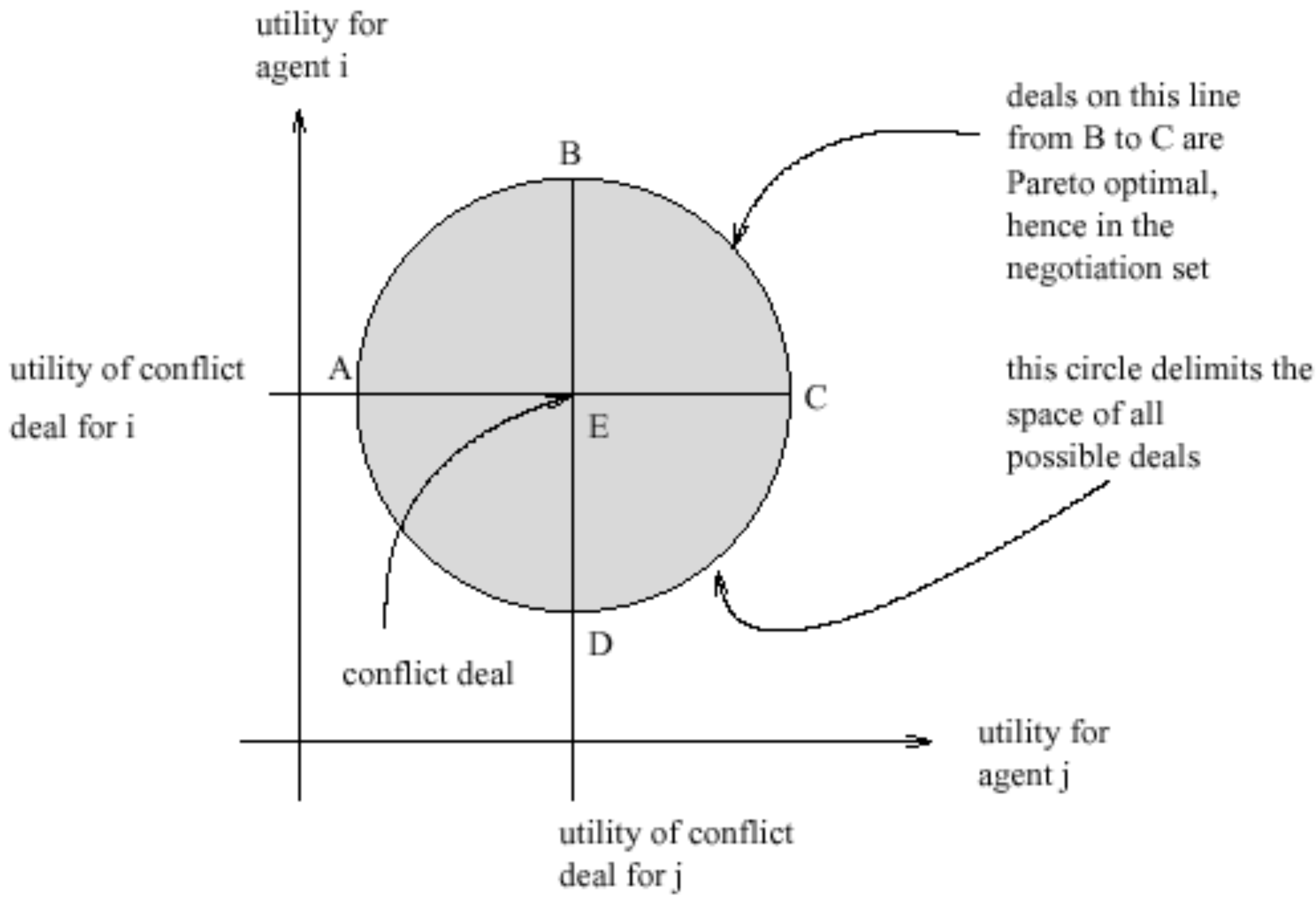






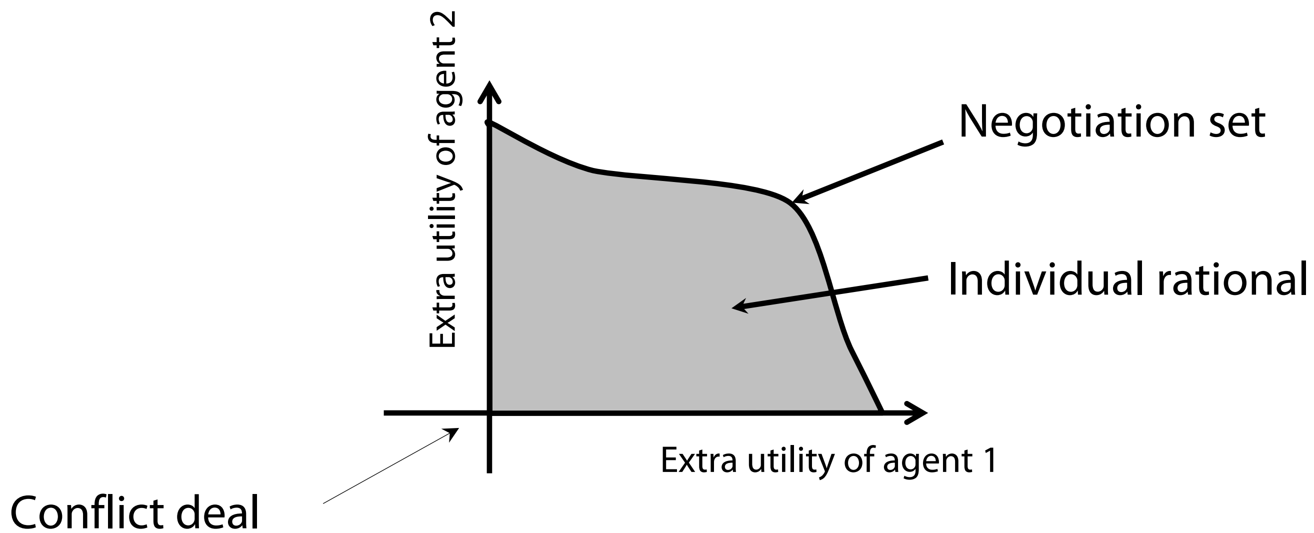
Cost is
only to
establish
connection

The Negotiation Set Illustrated

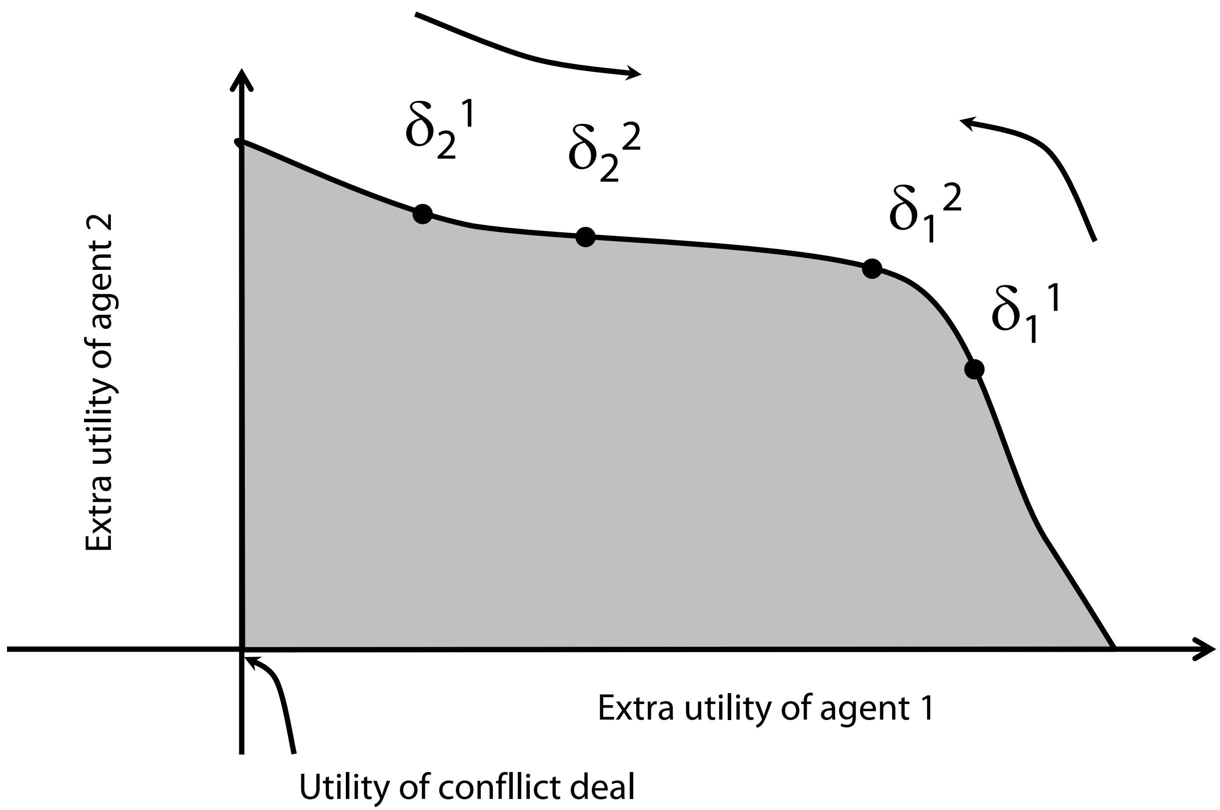


Negotiation Set

- The negotiation set consists of the deals that are Pareto efficient and individual rational
  - A deal is Pareto efficient (Pareto optimal) if it is not dominated by another task allocation
  - A deal is individual rational if neither agent is worse off than in the original allocation (the 'conflict deal')



Monotonic Concession Protocol



Monotonic Concession Protocol

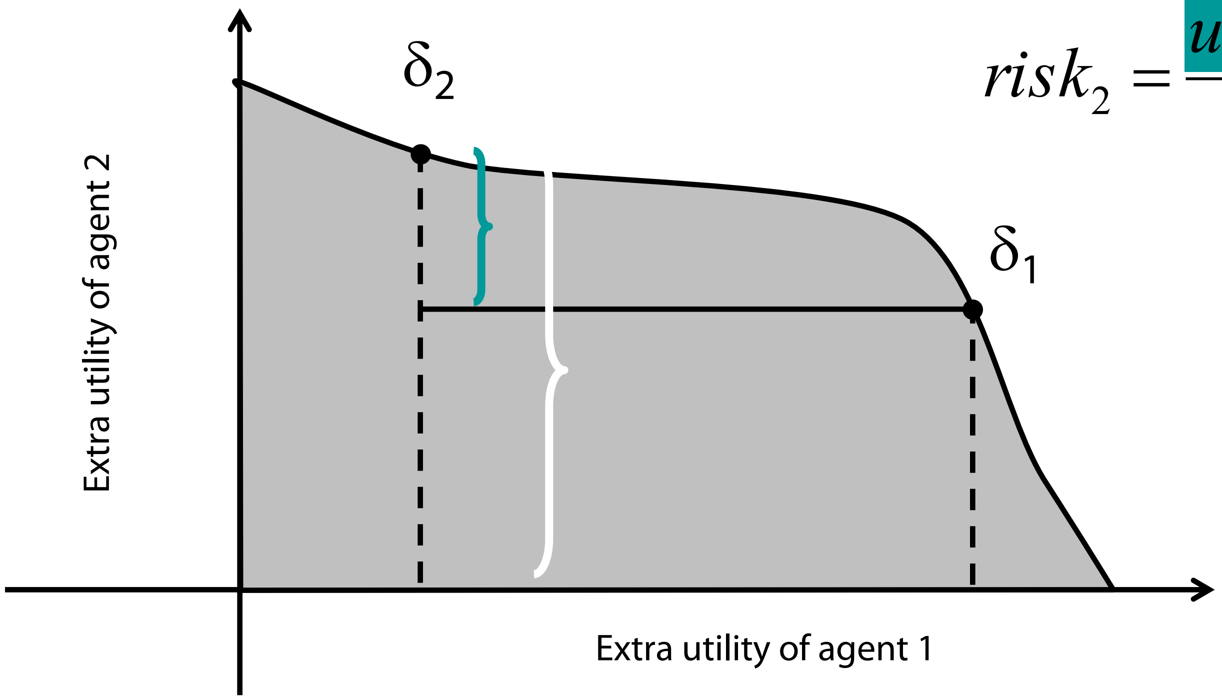
- Both agents make several small concessions until an agreement is reached.
  - Each agent proposes a deal
  - If one agent matches or exceeds what the other demands, the negotiation ends
  - Else, each agent makes a proposal that is equal or better for the other agent (concede)
- If no agent concedes, the negotiation ends with the conflict deal

Monotonic Concession Protocol

- What is a good negotiation strategy for the Monotonic Concession Protocol?
- Consider danger of getting it wrong:
  - If you concede too often (or too much), then you risk not getting the best possible deal for yourself.
  - If you do not concede often enough, then you risk conflict (which has utility 0).

Zeuthen Strategy

Idea: Analyse willingness to risk conflict



$$risk_2 = \frac{u_2(\delta_2) - u_2(\delta_1)}{u_2(\delta_2)}$$

The strategy was introduced in **1930** by the Danish economist Frederik Zeuthen.

Zeuthen Strategy

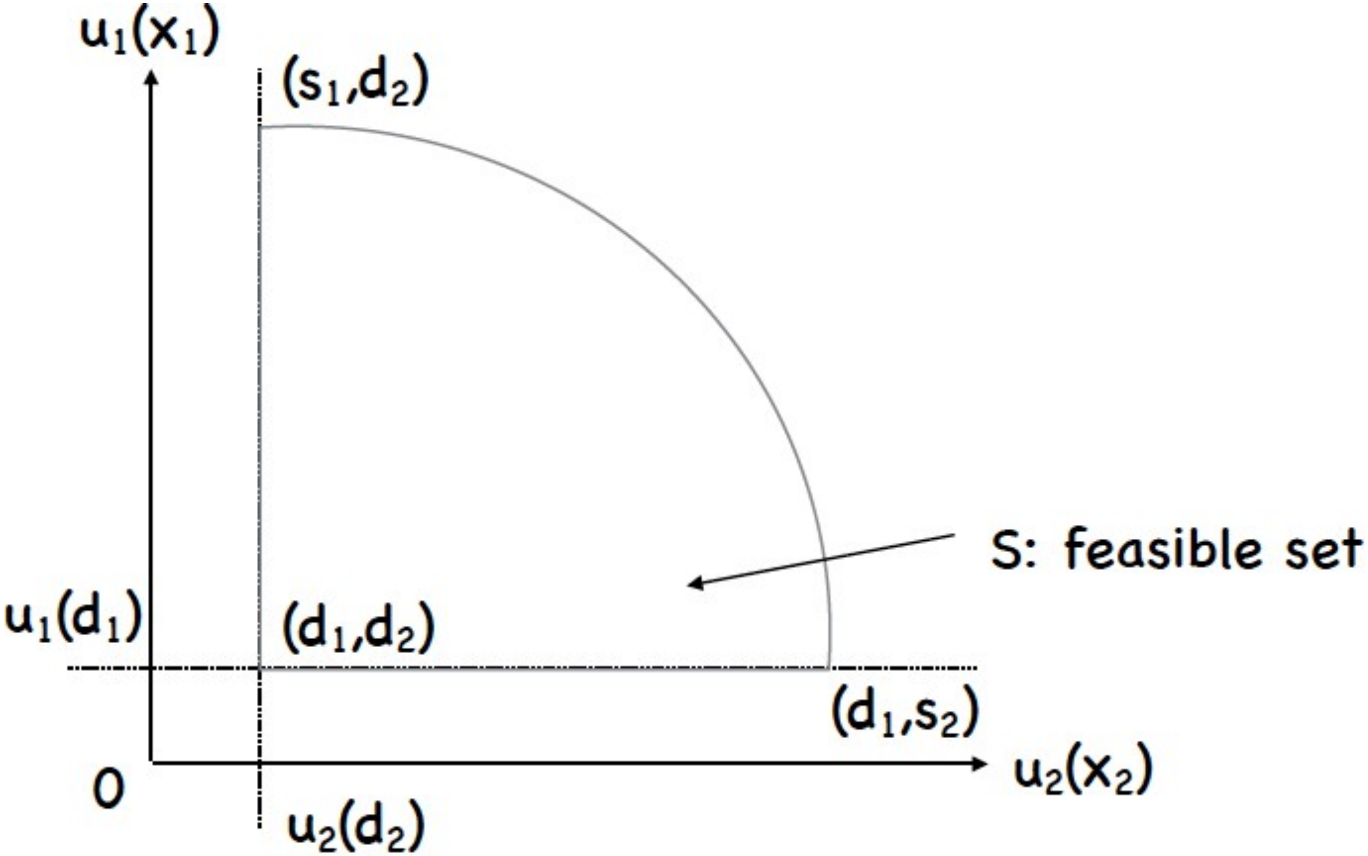
- Start with deal that is best among all deals in the negotiation space for yourself
- Calculate willingness to risk conflict of self and opponent
- If willingness to risk conflict is smaller than opponent, offer minimal sufficient concession (a sufficient concession makes opponent's willingness to risk conflict less than yours); else offer original deal

Game-Theoretic Analysis

- A bargaining problem is defined as a pair (S, d)
  - S is bargaining set, that is the set of all utility pairs resulting from an agreement
  - d is the disagreement point where each agent i gets $u_i(d)$ even if there is no agreement
- A bargaining solution is a function f that maps every barging problem (S, d) to an outcome in S, i.e., f(S,d)-> S
- A Nash solution is defined as follows:

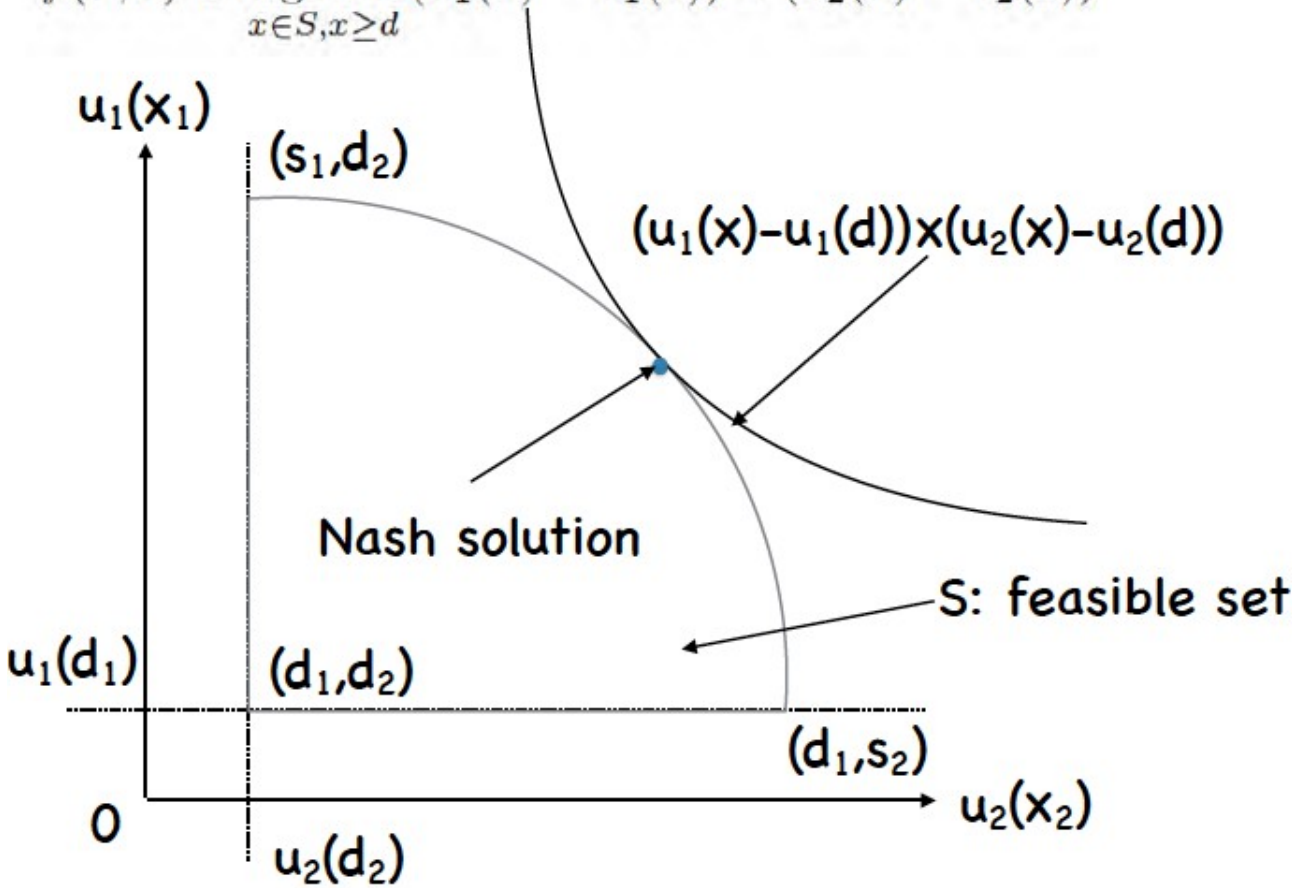
$$f(S,d) \in \arg\max_{x \in S, x \geq d} \left(\boxed{u_1(x) - u_1(d)) \times (u_2(x) - u_2(d)} \right)$$

Nash product



* It is assume that S is closed and convex.

$$f(S,d) \in \arg\max_{x \in S, x \geq d} (u_1(x) - u_1(d)) \times (u_2(x) - u_2(d))$$



* It is assume that S is closed and convex.

Nash Solution

- Nash proved that the solution that satisfies the five axioms below is a (unique) Nash solution:
  - Axiom 1 (Individual Rationality) : Each agent can get at least disagreement point: $f(S,d) \geq d$
  - Axiom 2 (Symmetry) : The solution is independent form agent's name, like A or B
  - Axiom 3 (Pareto Optimality)
  - Axiom 4 (Invariance from Affine Transformation) : The solution should not change as a result of linear changes to the utility for either agent
  - Axiom 5 (Independence of Irrelevant Alternatives) : Eliminating feasible alternatives that are not chosen should not affect the solution. Namely,

$$S' \subset S \text{ and } f(S,d) \in S', \text{ then } f(S',d) = f(S,d).$$

Negotiation Protocols

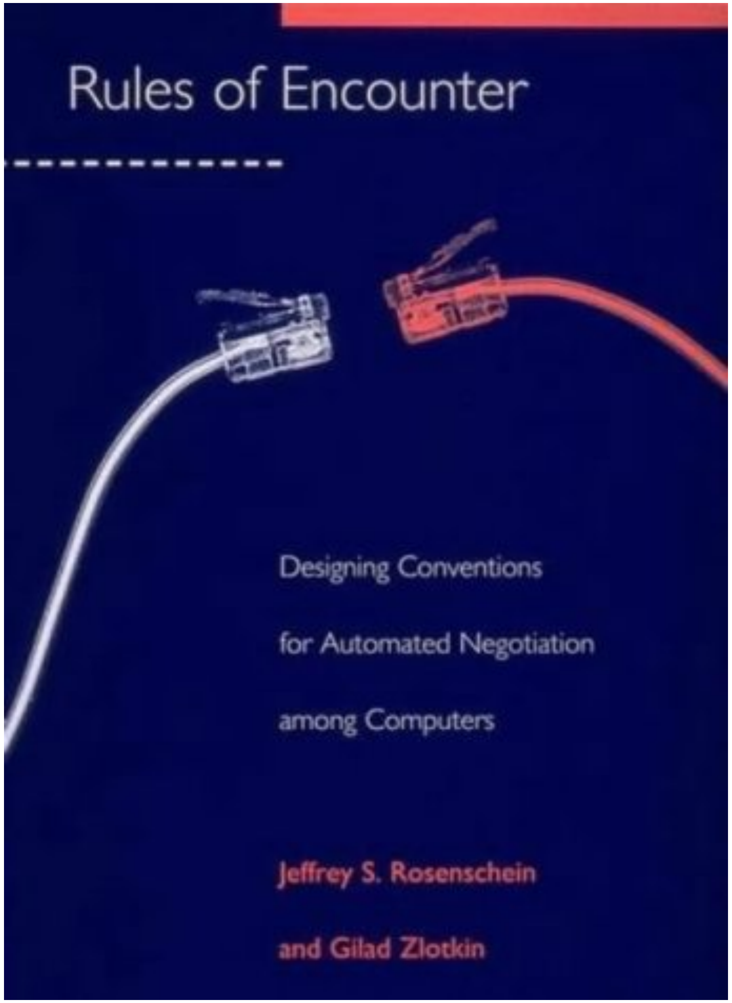
- Agents use a product-maximizing negotiation protocol
- It should be a symmetric PMM (product maximizing mechanism)
- Examples
  - 1-step protocol (e.g., second price sealed bid)
  - Monotonic concession protocol

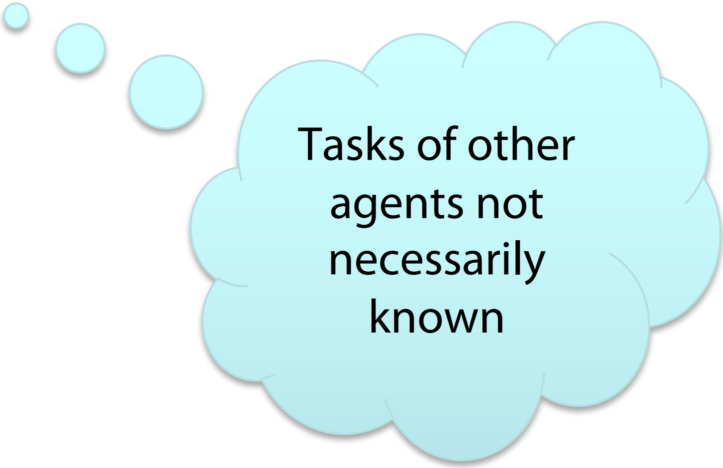
Summary: Monotonic Concession Protocol

- Properties
  - Termination: guaranteed if the agreement space is finite
  - Verifiability: easy to check that an opponent really concedes (only one's own utility function matters)

- Criticism
  - You need to know your opponent's utility function to be able to concede (typical assumption in game theory; not always appropriate)

Automated Negotiation among Agents: TOD

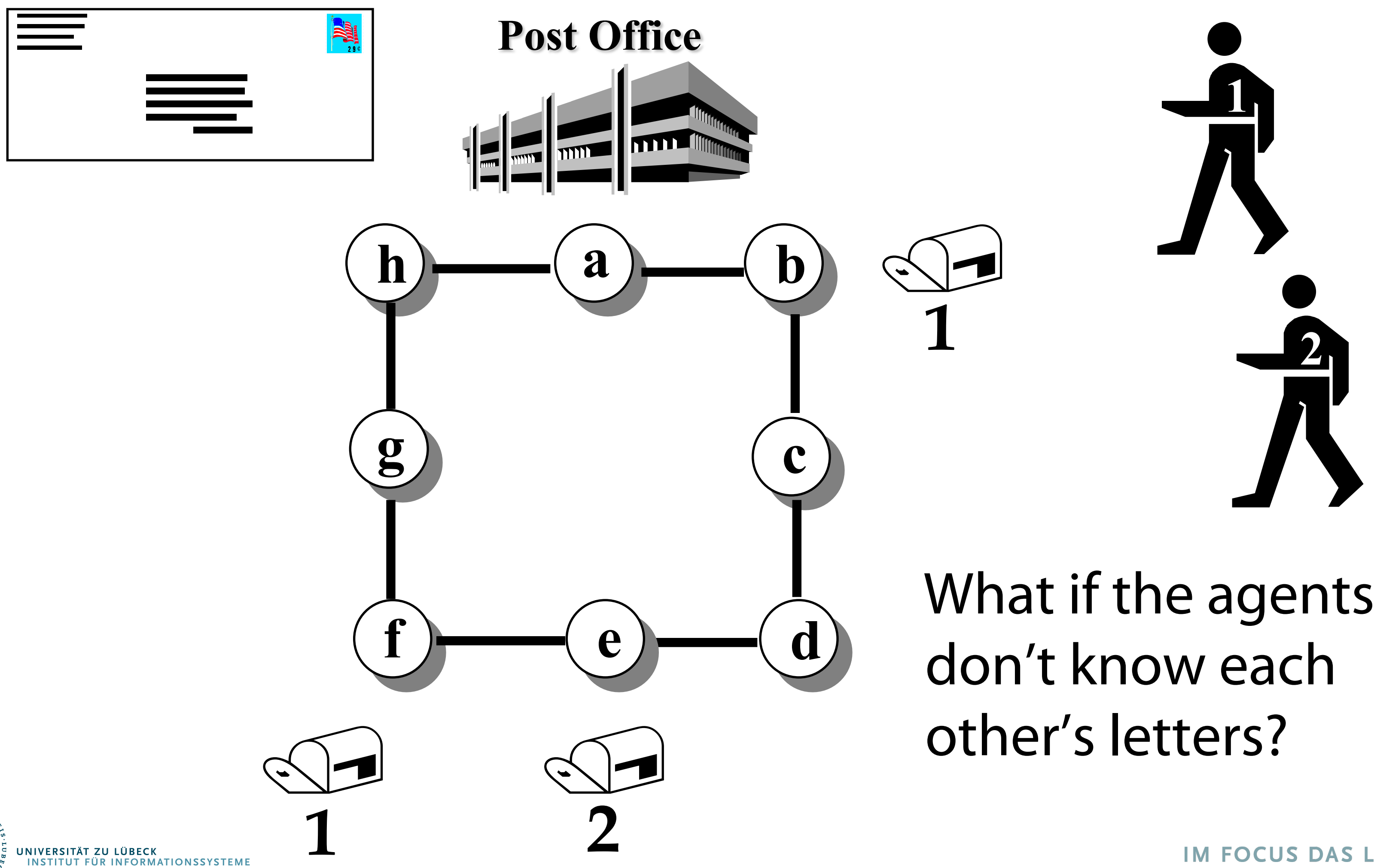




Jeffrey S. Rosenschein and Gilad Zlotkin. Rules of encounter: Designing Conventions for Automated Negotiation among Computers. MIT Press. **1994**.

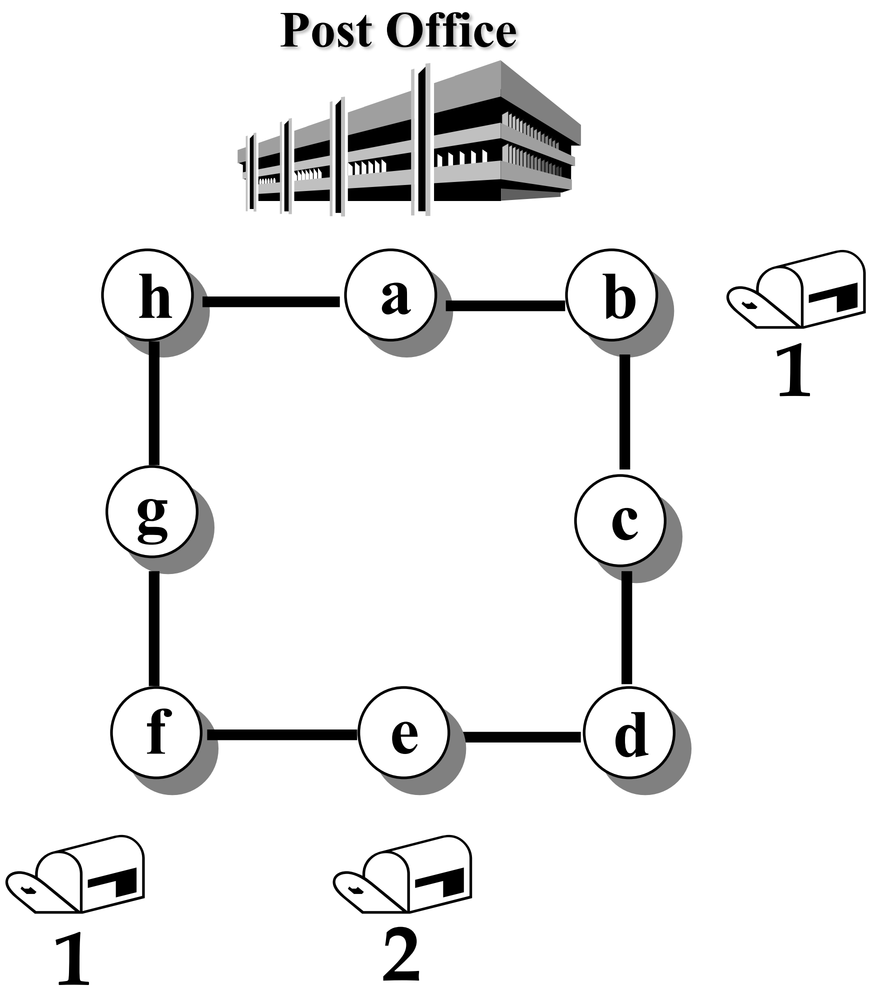
Jeffrey S. Rosenschein and Gilad Zlotkin
Designing Conventions for Automated Negotiation
AI Magazine Volume 15 Number 3. **1994**.

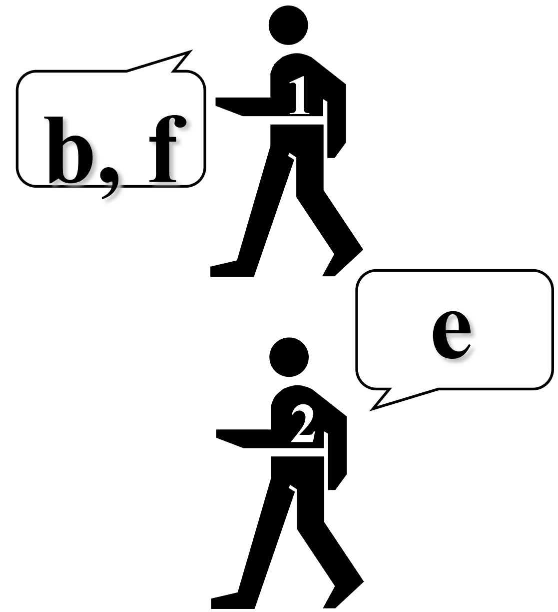
Negotiation with Incomplete Information



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

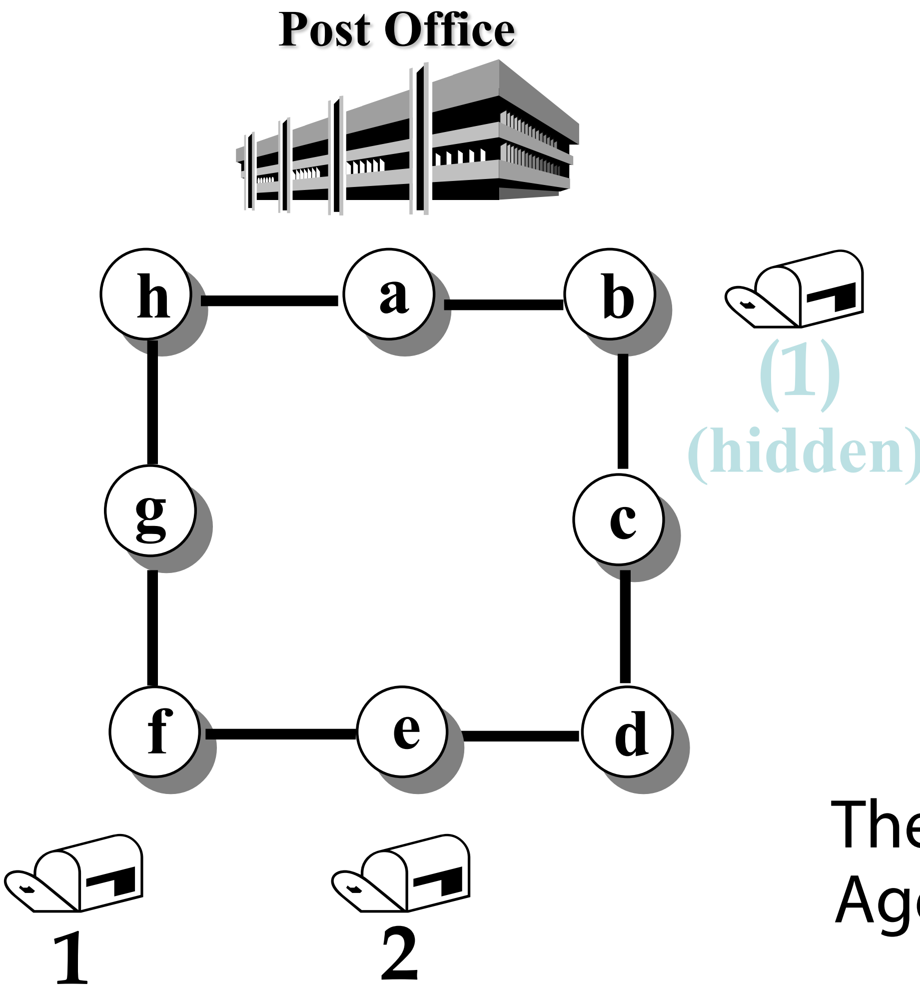
Pre-Game: Broadcast Tasks

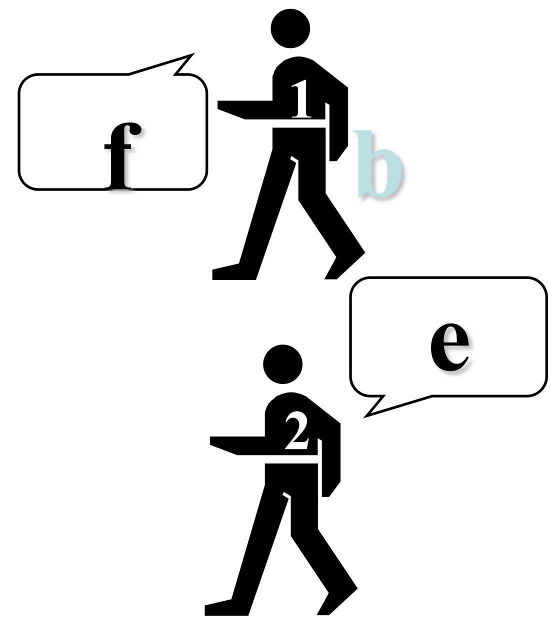




Agents will flip a coin to decide who delivers all the letters

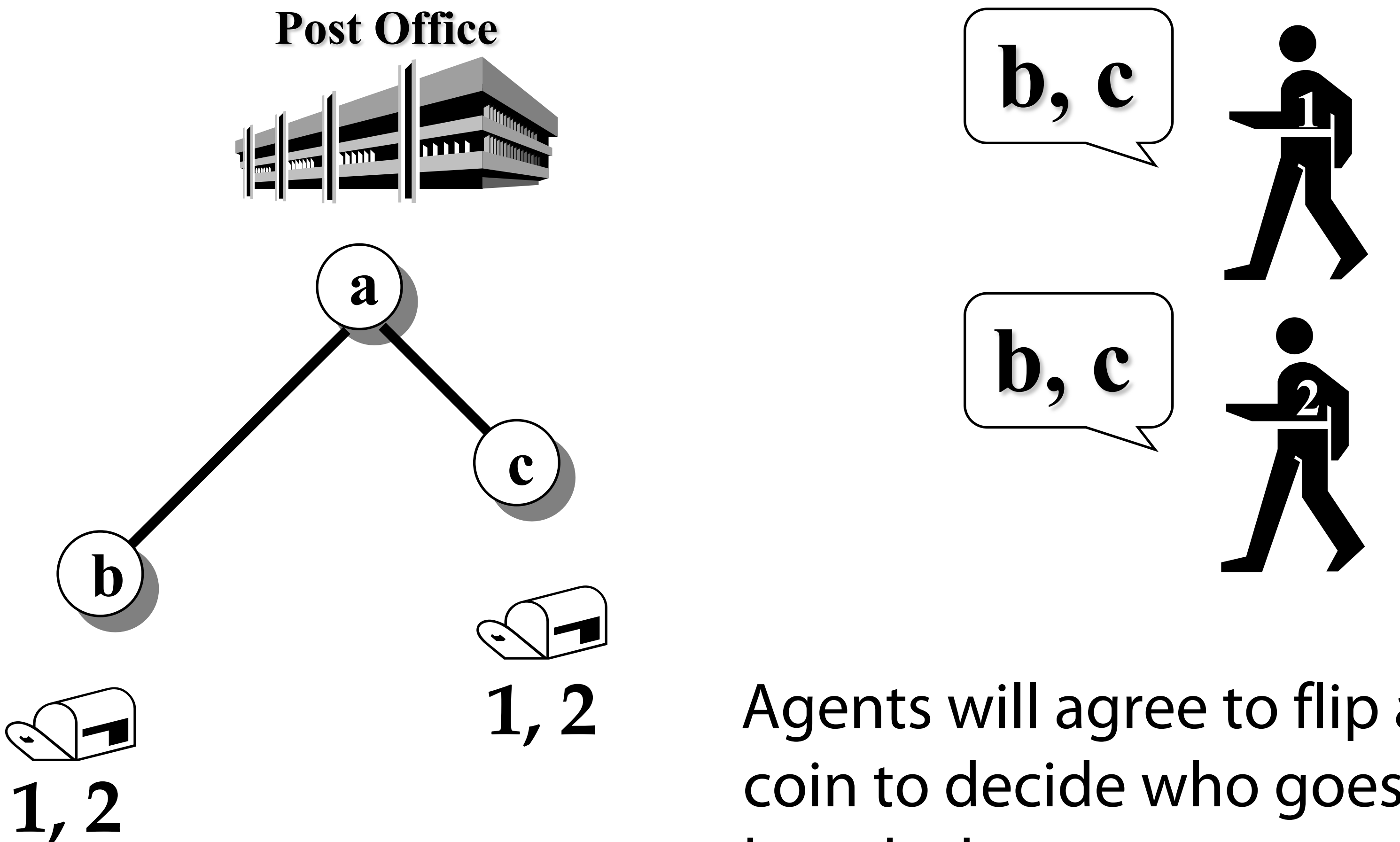
Hiding Letters





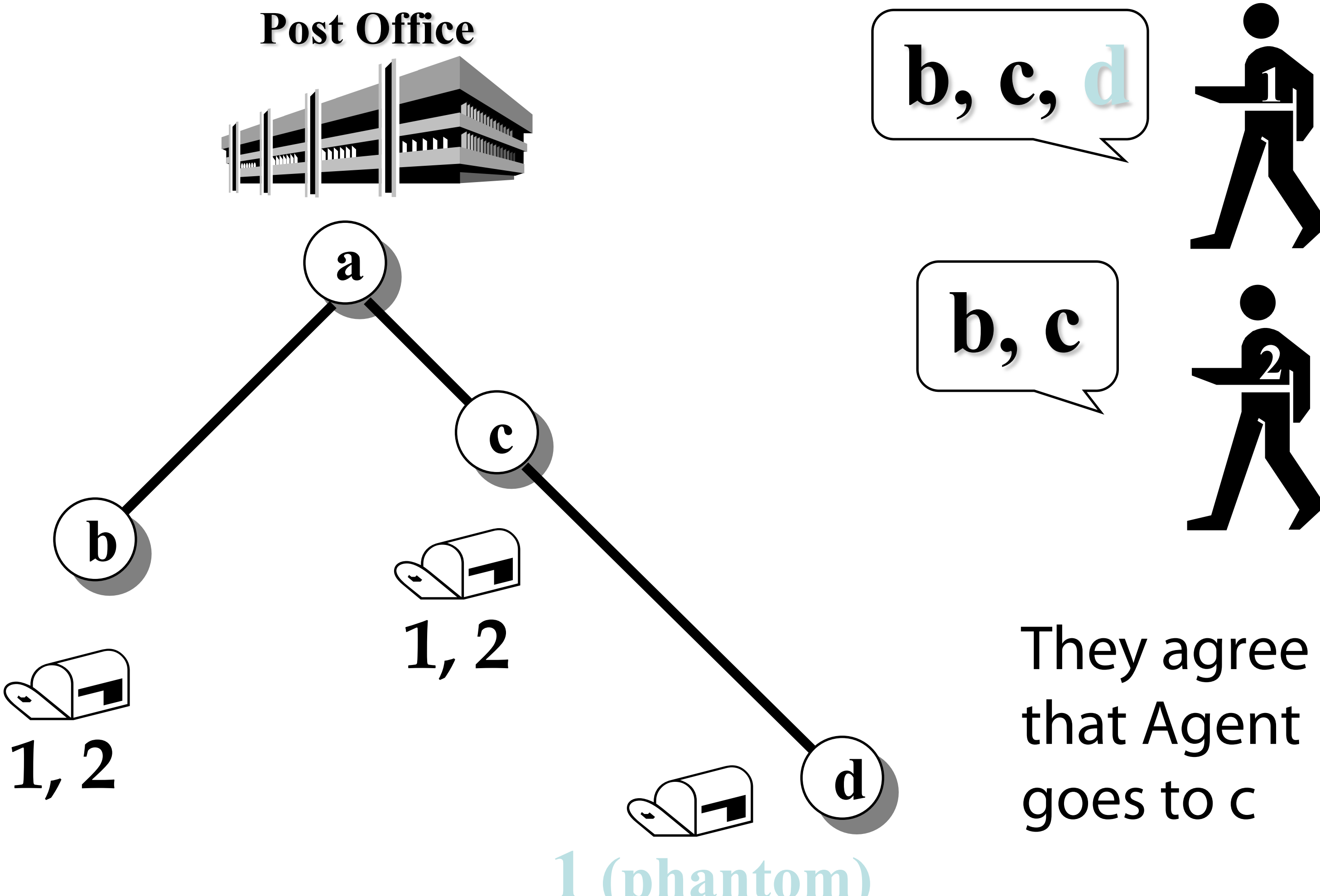
They then agree that Agent 2 delivers to f and e

Another Possibility for Deception



Agents 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 $<D_1, D_2> : p$

Agent 1 will perform D_1 with probability p and D_2 with probability 1-p, and vice versa.

Agents need to reason about expected utility

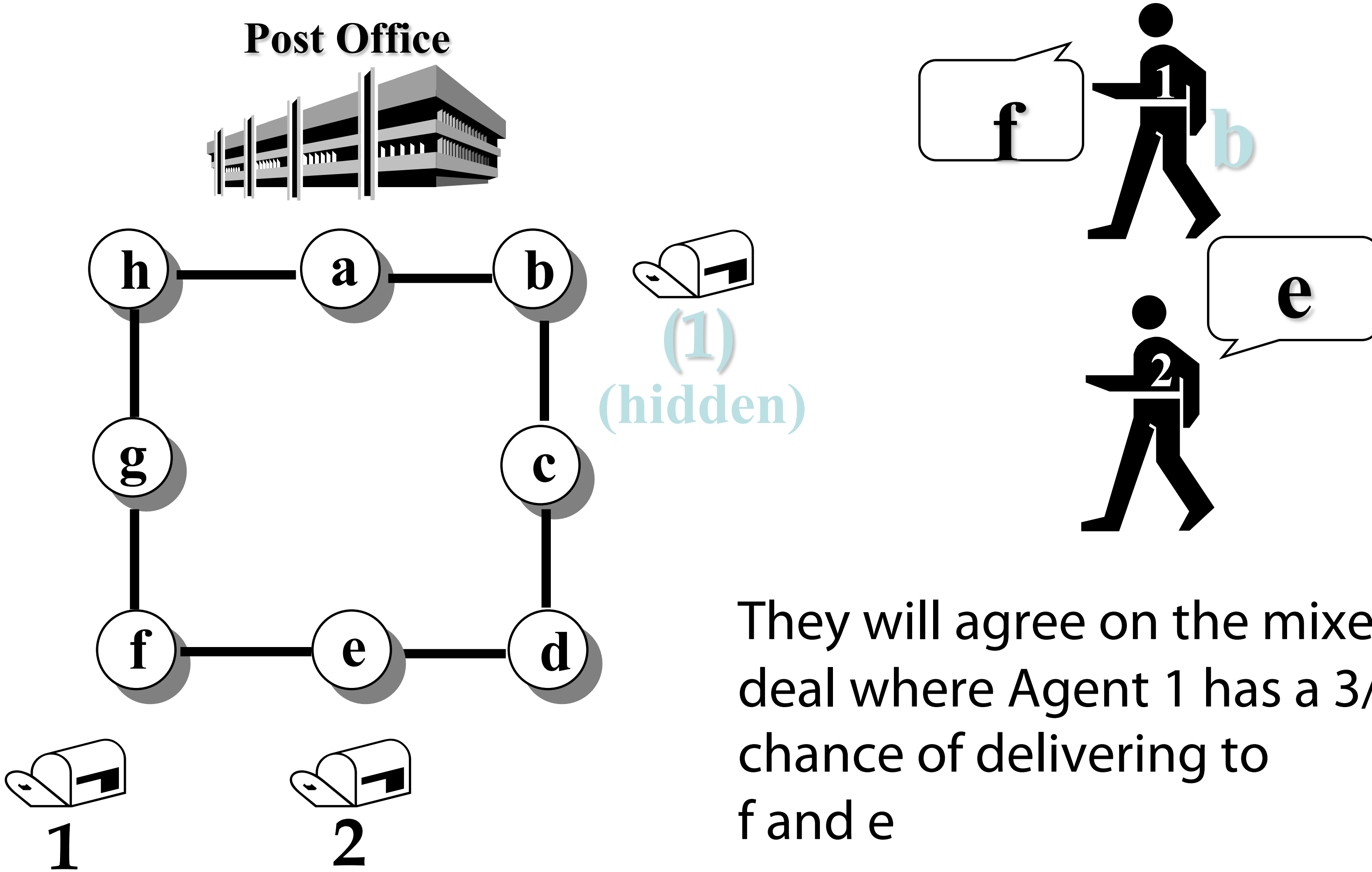
The more burden initially, the more burden in the final deal

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, and $p := 0.5 \mid T'_1 \mid / \mid T'_1 \cup T'_2 \mid$ where T_i’ is the announced task of agent i and |.| denotes the #steps

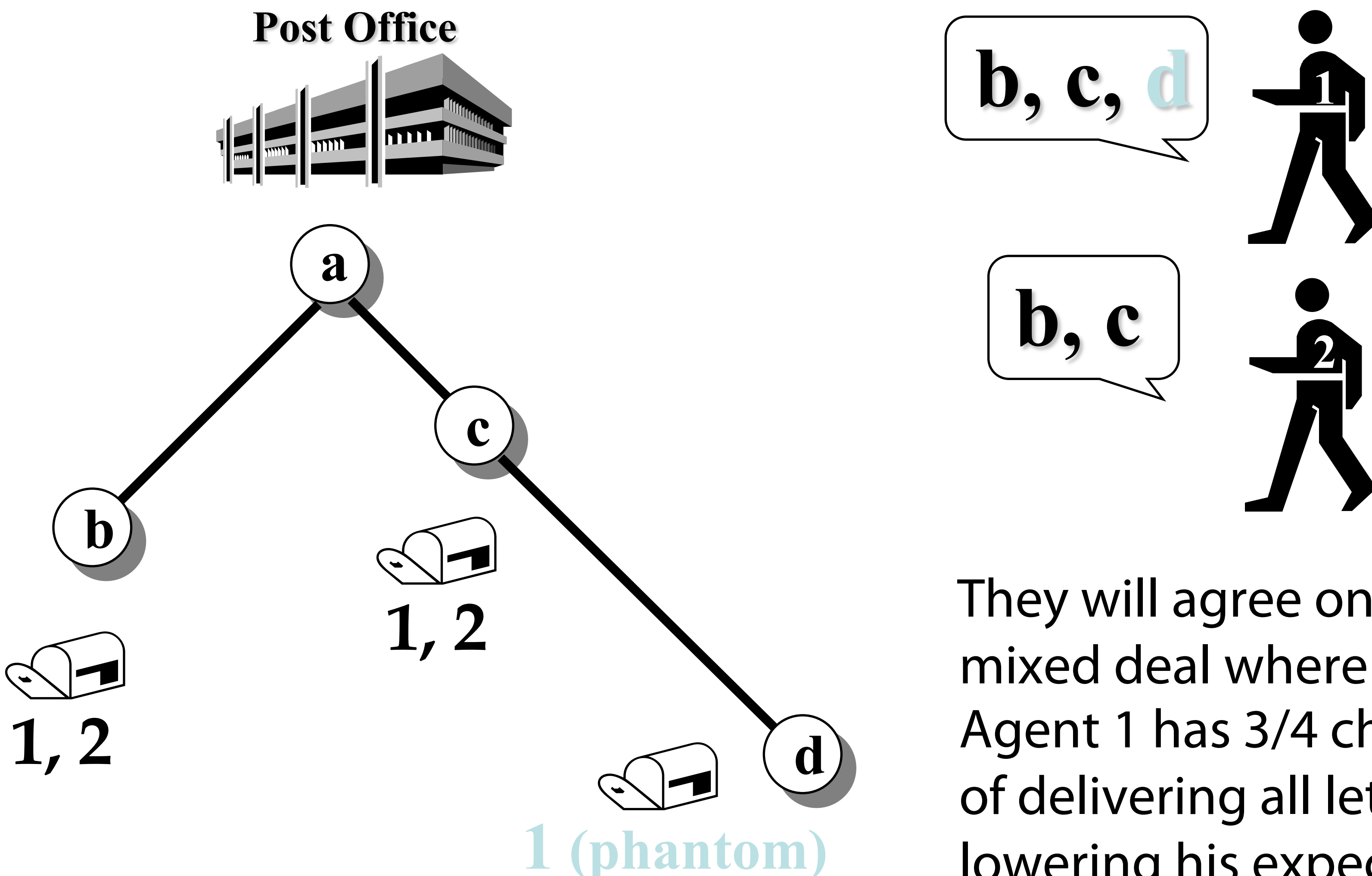
By adding probabilities, the deal space has become continuous

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



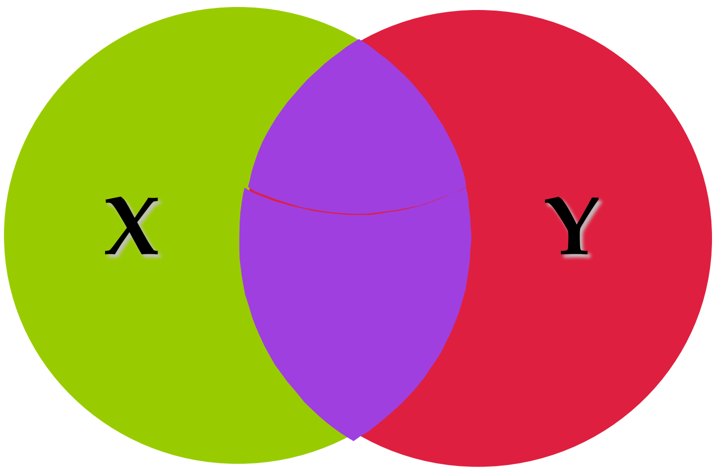
They will agree on the mixed deal where Agent 1 has 3/4 chance of delivering all letters, lowering his expected utility

Sub-Additive TODs

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

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

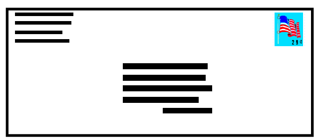
Sub-Additivity

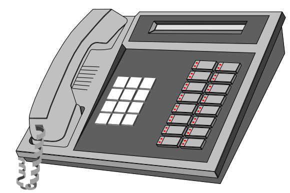


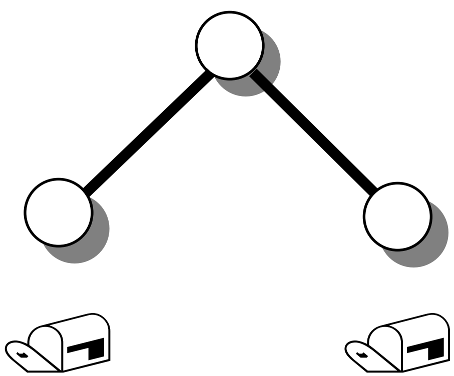
$$c(X \cup Y) \leq c(X) + c(Y)$$

Sub-Additive TODs

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







“Delivery Domain” (where postmen don’t have to return to the Post Office) is not sub-additive

Incentive Compatible Mechanisms

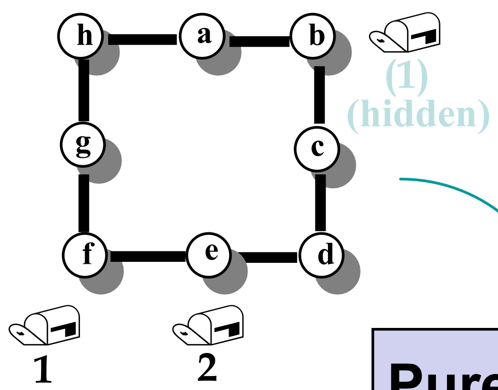
Sub-Additive

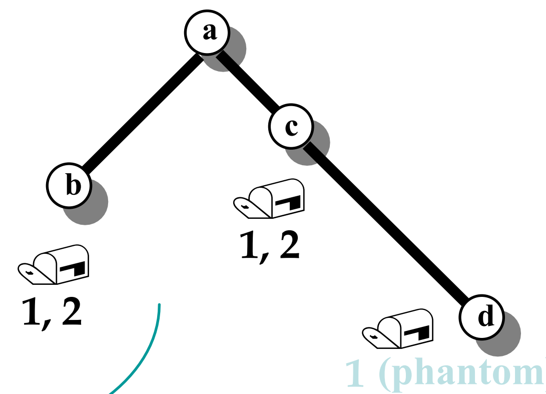
	Hidden	Phantom
Pure	L	L
A/N	T	T/P
Mix	L	T/P

A mechanism is called incentive-compatible (IC) if every participant can achieve the best outcome to themselves just by acting according to their true preferences

- 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.

product-maximizing mechanism (PMM)

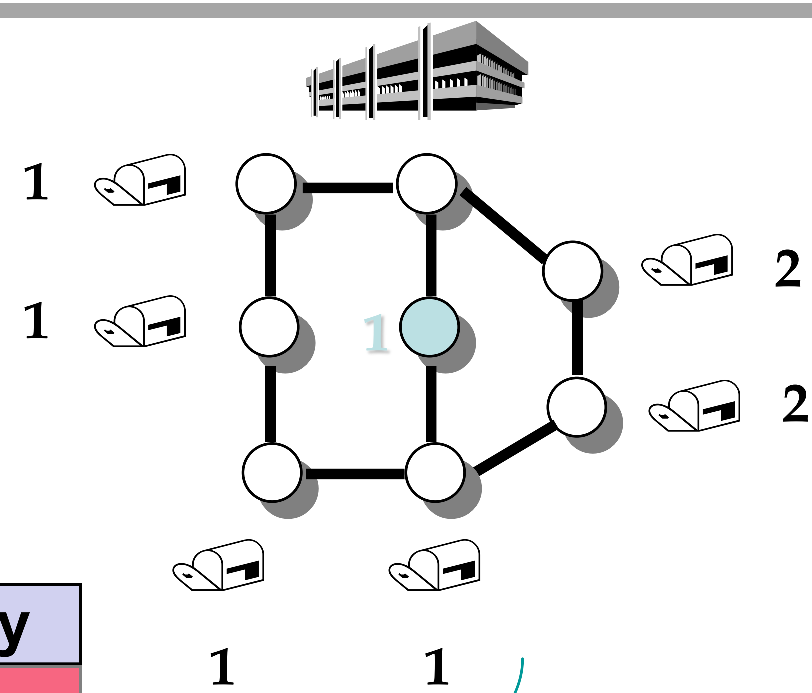
Incentive Compatible Mechanisms

	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)

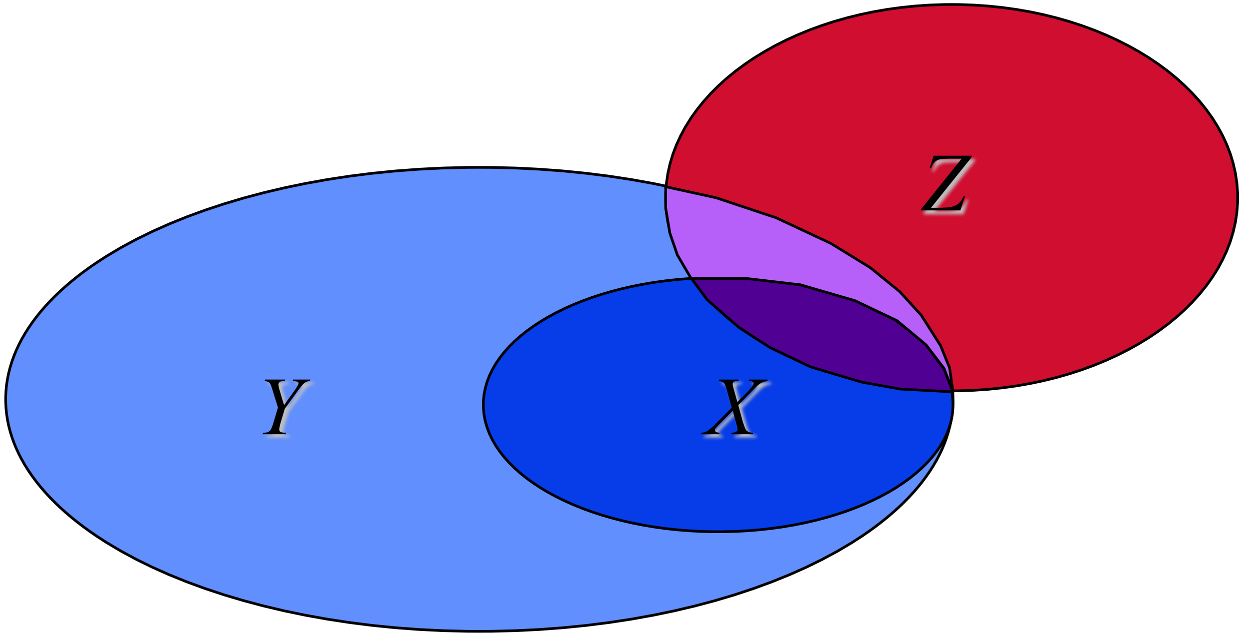
Concave TODs

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

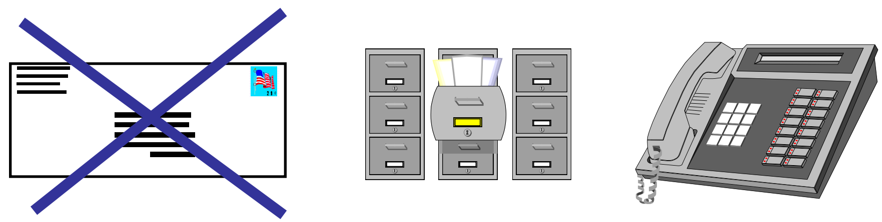
Concavity

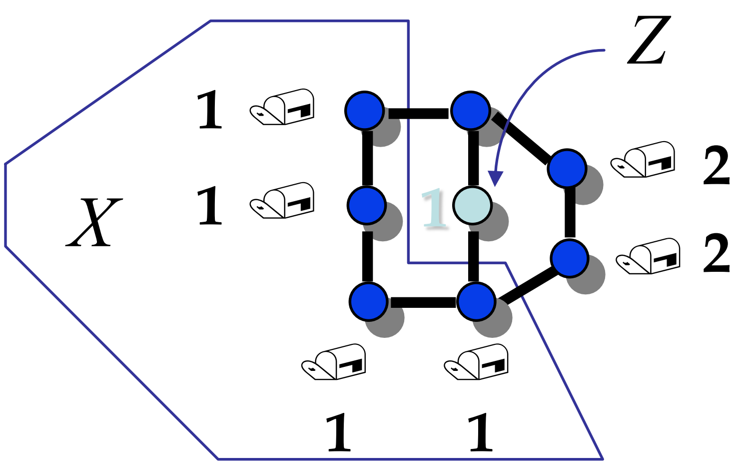


The cost Z adds to X is more than the cost it adds to Y.
(Z - X is a superset of Z - Y)

Concave TODs

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





This example is 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

product-maximizing mechanism (PMM)

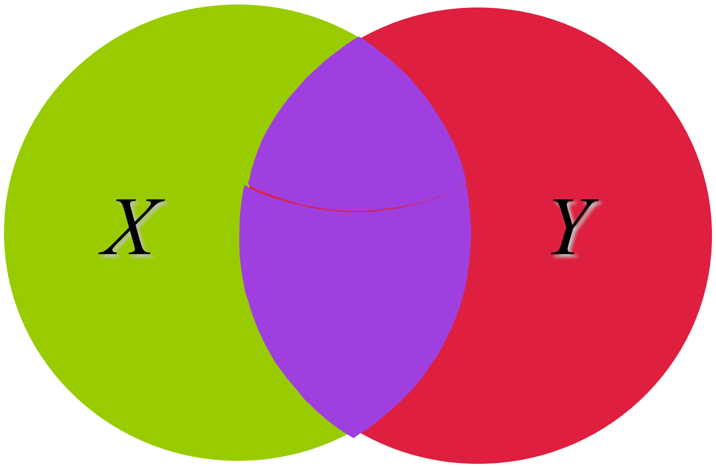
Modular TODs

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

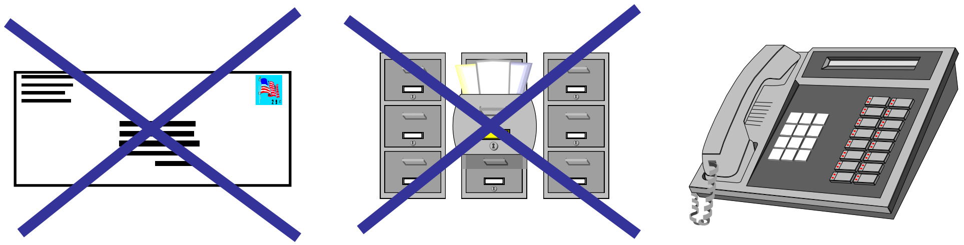
Modularity



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

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

Modular

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

Concave

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

Sub-Additive

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

Related Work

- Similar analysis made of State Oriented Domains, but the 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)

Summary

- By appropriately adjusting the *rules of encounter* by which agents must interact, we can influence the private strategies that designers build into their machines
- The interaction mechanism should ensure the *efficiency* of multi-agent systems

Rules of Encounter

Efficiency

Summary

- 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

$\boxed{\text{Stability}}$

$\boxed{\text{Formal Tools}}$