Distributed AI

Welcome!

- **Past exams (warning, hot!)**
- Sample exam 1 | Sample 2 | Sample 3

Books

- An Introduction to MultiAgent Systems, Woolridge, 2nd Edition
- Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence (covers lots of material)
- An Introduction to Multi-Agent Systems (MAS, Nash equilibrium, KQML)
- Distributed Rational Decision Making (social welfare, Pareto efficiency, social rule)
- Consenting Agents: Designing Conventions for Automated Negotiation (task-oriented domains)
- Cute website

How the exam works

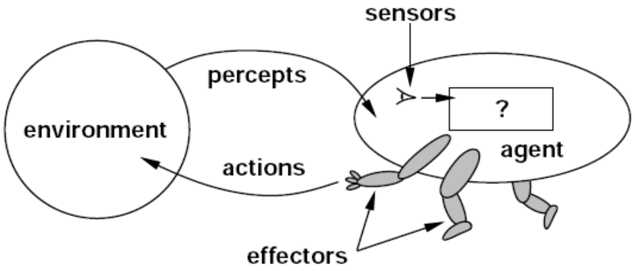
- There are three categories of questions in the exam
- Last year part one had 12 questions
- You must pass enough number of question in the first category (most important category) to pass

Good luck!

1. What is an agent?

Agents

An agent is an entity that acts on behalf of another entity or a person and has a decision making capability/power. The agent takes sensory input from the environment, and produces output actions that affect it. The interaction is usually an ongoing, non-terminating one.



What are the main differences between agents and objects?

The differences between agents and objects are

1. Higher degree of autonomy: agents execute independently. Objects do not have control of their methods and only encapsulate state whereas agents encapsulate behavior.
2. Higher degree of smartness: agents have properties of reactivity, proactiveness, and social ability.
3. Higher degree of activeness: agents are generally multi-threaded.

Explain the difference between cooperative and self-interested agents. Give examples. How can this difference be expressed in terms of a utility function?

- Cooperative agents work to maximize the utility of the whole system, and their preferences come secondary. Example: dancers in a choreography
- Self-interested agents follow their own agenda first, and the system's preferences are secondary. Example: bidders

What are the proposed weak and strong properties of agents? Briefly explain them and give an example for each.

Weak properties:

1. Autonomy: execute independently without human intervention
2. Proactiveness: can take initiative
3. Reactivity: act with respect to input from the environment
4. Social ability: communicate with other agents

Strong properties:

5. Benevolence: having no conflicting goals, doing what it is being told
6. Rationality: achieving the given goals
7. Veracity: communicate truthful information
8. Mobility: move around in the network

Want to remember them?
weak properties **APRS (A**verage **P**rofit **R**ate**S)**
strong properties **BRVM (BR**azil **V**irtual **M**achine**)**

To cite an example, simply mention one of the answers below.

The availability of TV channels will be greatly extended in the information society. Media experts forecast that shortly after the new millenium, every household will be able to receive between 300 and 500 television programs. A quick estimate shows that a systematic search for interesting TV programs would take so much time that there would be hardly any time left to view the selected broadcasts. Assume that you are supposed to help in solving this problem and to create an intelligent agent for that. List all possible features the intelligent TV agent will have and illustrate these features by examples.

The user informs the agent of his TV preferences, the agent searches the available digital TV programs on behalf of the user and then informs the user what will be shown on TV concerning his/her preferences.

Weak properties:

1. Autonomy: the agent continuously follows new broadcasts and television programs, and provides the user with new information on his areas of interest.
2. Proactiveness: the agent may inform the user about broadcasts that the user may not express interest in, but that the agent thinks the user could find equally interesting.
3. Reactivity: the agent can react to changes in the TV schedule.
4. Social ability: the agent can communicate with other TV agents and ask for their opinion about some programs or ask for help.

Strong properties:

5. Benevolence: the agent will always try to do what is asked of it.
6. Rationality: the agent will follow the user's preferences as best as it can.
7. Veracity: the agent will not reply with false information.
8. Mobility: the agent can move from one computer to another when traversing the TV programs.

Analyze an elevator (lift) from an agent's perspective. A simple elevator is a small room that carries people from one floor to another. It can be assumed to have two doors, a display indicating the floor it's on, buttons on the outside for the user to call it, and buttons on the inside for the user to indicate the floor that he wants to move to. Assume that the elevators are intelligent agents. List all possible agent properties the intelligent elevator agents will have and illustrate these properties by examples.

Weak properties:

1. Autonomy: once the user indicates the floor number the elevator moves up to the desired floor without requiring human intervention at each floor.
2. Proactiveness: the elevator updates the display number proactively once it crosses each floor. （This one looks like autonomy. What about doors will not shut down if someone is still passing throungh the door ）
3. Reactivity: while the elevator is moving up, it will stop at any floor if a user presses a button outside at that floor.
4. Social ability: the elevator might communicate with other elevators to divide the task of going up and going down.

Strong properties:

5. Benevolence: the elevator will move to the desired floor without any conflicting goals.
6. Rationality: the elevator will not act in any way to prevent its goals from being achieved.
7. Veracity: the display will indicate the correct floor number.
8. Mobility: the elevator moves up or down depending on the direction indicated by the user.

If traffic lights (together with their control systems) are to be considered as intelligent agents, which of all the possible agent's properties will they employ and which properties they will not employ? Illustrate your answer by examples.

Properties traffic lights employ:

1. **Autonomy: the traffic light switches lights without human intervention.**
2. **Proactiveness: the traffic light will switch from red to green if many cars are waiting.**
3. **Reactivity: when a person presses the button, the traffic light will switch to green sooner to let the person cross.**
4. **Social ability: traffic light agents might communicate with other traffic lights to coordinate the traffic.**
5. **Benevolence: the goal of traffic lights is to control the traffic and they do so.**
6. **Rationality: the traffic light won't make traffic conditions worse.**
7. **Veracity: the traffic light will not give false information and always shows the right color.**

Properties they can't employ:

8. **Mobility: traffic lights are fixed and they cannot move.**

Let us consider a control system that regulates the intensity of light in an auditorium as an intelligent agent. Explain which intelligent agent properties this agent may employ. Illustrate your answer with examples.

Properties it will employ:

1. **Autonomy: the control system will automatically switch on and off the lights without human intervention if there's at least one person.**
2. **Proactiveness: the control system will turn the lights off to save energy if there's enough light coming in from the windows.**
3. **Reactivity: the control system will switch off the lights if the projector is on.**
4. **Social ability: the control system will communicate with other agents to coordinate the power consumption in the building.**
5. **Benevolence: the control system will do what is being asked, with no conflicting goals.**
6. **Rationality: the control system will make sure the light is optimal and won't want to waste energy.**
7. **Veracity: the control system will always display the correct light intensity in the companion screen.**

Properties it doesn't employ:

8. **Mobility: the control system is fixed and cannot move.**

Distributed Artificial Intelligence

The primary focus of research in the field of DAI has included three different areas: Distributed problem solving (DPS), Multi-agent systems (MAS), Parallel AI.

Explain the difference between Distributed Problem Solving and Multi-Agent Systems.

- Distributed Problem Solving (DPS) divides a problem into subproblems and allocates them to entities. Centralized, more predictable, less robust.
- In Multi-Agent Systems individual agents share knowledge and communicate with each other to solve a problem that is beyond the scope of a single agent. Decentralized, less predictable, more robust.

2a. Agent Negotiation

Negotiation Principles

Explain what is utility and what is preference and how they are related?

- Preferences are the agents' preferred outcomes.
- Utility is a numerical representation of a preference and agents use it to compare the outcomes to identify which one is best.

What are the components of any negotiation setting? Explain them.

1. Negotiation set: the set of all possible proposals an agent can make
2. Negotiation protocol: the rules of negotiation
3. Strategies: which proposals an agent will make
4. Rule: determines when an agreement is reached

Negotiation Protocols

Assume that τ denotes a state transformer function whose the first parameter is action of Agent1 and the second argument is action of Agent2. How do you characterize the following environments?

1) $\tau(D,D) = \omega 1 \quad \tau(D,C) = \omega 1 \quad \tau(C,D) = \omega 2 \quad \tau(C,C) = \omega 2$

2) $\tau(D,D) = \omega 1 \quad \tau(D,C) = \omega 2 \quad \tau(C,D) = \omega 3 \quad \tau(C,C) = \omega 4$

3) $\tau(D,D) = \omega 1 \quad \tau(D,C) = \omega 1 \quad \tau(C,D) = \omega 1 \quad \tau(C,C) = \omega 1$

4) $\tau(D,D) = \omega 1 \quad \tau(D,C) = \omega 2 \quad \tau(C,D) = \omega 1 \quad \tau(C,C) = \omega 2$

1. environment controlled by Agent1
2. environment sensitive to actions of both agents
3. environment where neither agent has any influence
4. environment controlled by Agent2

What are the desirable properties of negotiation protocols? Briefly explain them.

1. Guarantee of success: the negotiation protocol should eventually terminate.

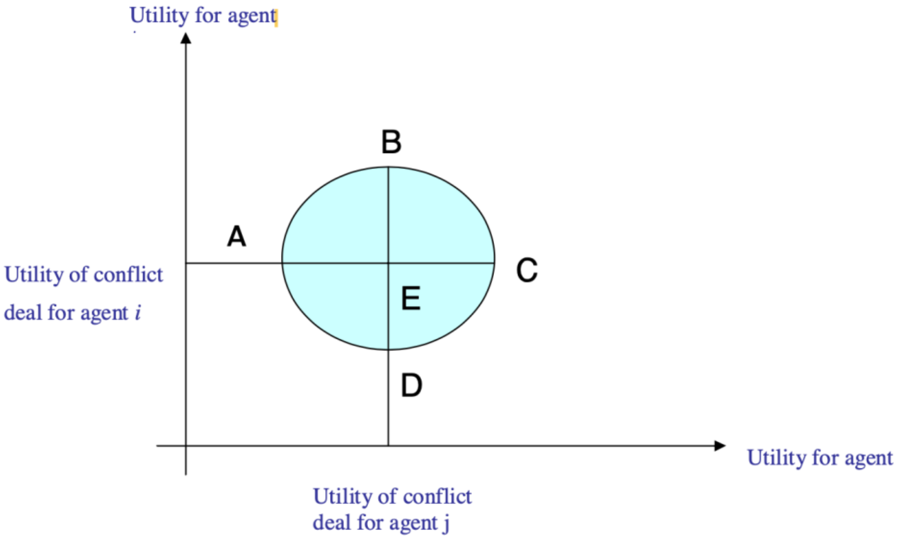
2. Maximizing social welfare: sum of all agents' utilities.
3. Pareto optimality: when an agent's utility can be increased if and only if another agent's utility decreases.
4. Nash equilibrium: because neither agent has any incentive to deviate from a Nash equilibrium, thus agents choose their strategies easily.
5. Individual rationality: agents negotiate only if it makes individual rational sense to do so.
6. Computational efficiency: the negotiation should not involve too many computations.
7. Distribution: all else being equal, distributed protocols should be preferred to avoid a single point of failure.
8. Stability: the protocol should be non-manipulable.
9. Fairness: order shouldn't matter: an agent shouldn't have an advantage if they start first or last.

Pareto Optimality

Pareto efficiency is an important property in negotiation. Briefly explain it. What are relations between pareto efficient solutions and social welfare maximizing solutions?

- Pareto efficiency is when an agent's utility or payoff can be increased if and only if another agent's utility or payoff decreases.
- Social welfare maximizing solutions are a subset of Pareto efficient solutions. In other words, if a solution maximizes social welfare, this solution must be Pareto efficient. However, if a solution is Pareto efficient it doesn't necessarily mean it maximizes social welfare.

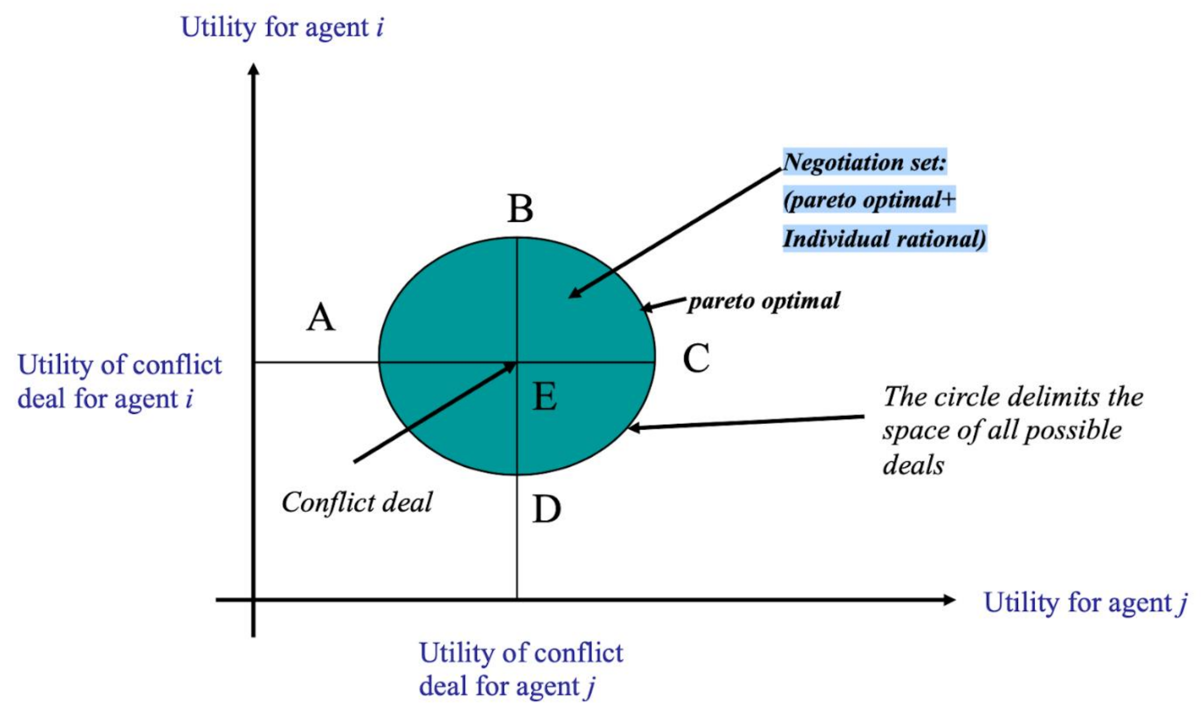
We considered the following figure in class:


Utility for agent
B
A
C
E
D
Utility of conflict deal for agent i
Utility for agent
Utility of conflict deal for agent j

Show on the figure the points/areas for: conflict deal, negotiation set, possible deals, Pareto optimal deals and individually rational deals.

- Conflict deal: point E

- **Negotiation set: BEC**
- **Possible deals: the entire circle**
- **Pareto optimal deals: BC curve**
- **Individually rational deals: semicircle ABCA for agent *i*, semicircle BCDB for agent *j***



Binary & Borda Protocols

Explain Condorcet's paradox. Give an example.

Suppose we have three outcomes, and three voters with the following preferences:

$$\omega_1 \succ_1 \omega_2 \succ_1 \omega_3$$

$$\omega_3 \succ_2 \omega_1 \succ_2 \omega_2$$

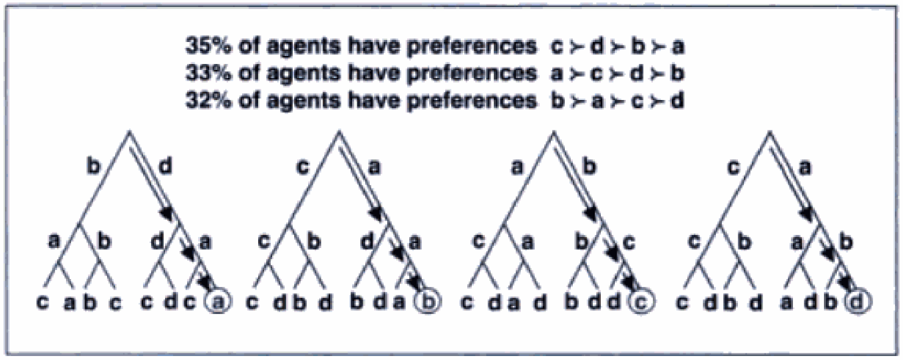
$$\omega_2 \succ_3 \omega_3 \succ_3 \omega_1$$

There is no winner, since each outcome is ranked first exactly once, and no matter which outcome we choose, the majority of voters will be unhappy.

Explain the binary protocol for voting. What are the possible outcomes of the binary protocol for the following set of preferences?

- **35% of agents have preferences c > d > b > a**
- **32% of agents have preferences b > a > c > d**
- **33% of agents have preferences a > c > d > b**

The idea of the binary protocol is that a pair of outcomes will face each other in a pairwise election, and the winner will then go on to the next election. The key problem is that the final outcome selected may depend not just on voter preferences, but on the order in which the candidates come up for election (i,e. the election agenda).



35% of agents have preferences $c \succ d \succ b \succ a$
33% of agents have preferences $a \succ c \succ d \succ b$
32% of agents have preferences $b \succ a \succ c \succ d$

Figure 5.1 Four different agendas for a binary protocol with four alternatives: a, b, c, and d.

If we compare the binary protocol and the Borda protocol, which of them is more computationally efficient? Which of them is not independent of irrelevant alternatives? Explain.

- **The Borda protocol is more efficient because the binary protocol uses pairwise comparisons which scale badly.**
- **The Borda protocol is not independent of irrelevant alternatives (adding or removing an alternative may completely change its outcome).**

Explain the Borda protocol. Which desirable property of a social choice rule is not satisfied in this protocol?

- **The Borda protocol assigns an alternative |O| points whenever it is highest in some agent's preference list, |O| - 1 whenever it is the second and so on. These counts are summed across voters, and the alternative with the highest count becomes the social choice.**
- **Properties not satisfied: transitivity and independence of irrelevant features**

Clarke Tax (strategy against insincere voters)

-- This is actually Clarke's tax, named after this guy, in practice, it is used in online advertising auctioning

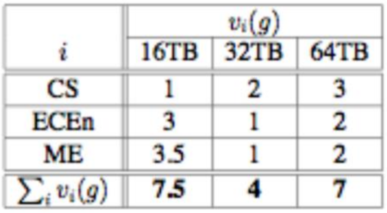
It seems like the Prof. keeps making the same typo and calls it Clark?

We considered the Clark tax algorithm. What is its objective? How does it achieve this objective? You don't have to present particular formulae in your answer.

Objective: deal with voters' cheating. Clark tax forces the voters that change the outcome of a vote to pay for the cost they impose on others that did not want that outcome. Thus, they will still have to pay, and then they might as well be truthful. By cheating the agent incurs a tax, the tax lowers the utility for the agent to such a

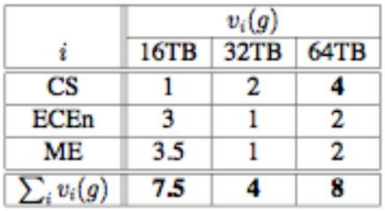
level that cheating and getting their prefered outcome has a lower utility than not cheating and getting their least prefered outcome. However, agents may still be able to cheat with a coalition.

We considered an example of application of Clark tax algorithm to the case when Computer Science (CS), Electrical and Computer Engineering (ECEn) and mechanical Engineering (ME) departments buying memory (see tables below). We assumed that a) represents truthful ranking and b) represents non truthful ranking. Explain using Clark tax algorithm why it is not profitable for the CS department to announce the non-truthful ranking?



i	$v_i(g)$		
	16TB	32TB	64TB
CS	1	2	3
ECEn	3	1	2
ME	3.5	1	2
$\sum_i v_i(g)$	7.5	4	7

Table 1. Truthful ranking



i	$v_i(g)$		
	16TB	32TB	64TB
CS	1	2	4
ECEn	3	1	2
ME	3.5	1	2
$\sum_i v_i(g)$	7.5	4	8

Table 2. Non-truthful ranking

It is not profitable for the CS department to announce the non-truthful ranking because by lying about its preferences it has to pay a tax (2.5) which exceeds the amount of benefit that CS acquired by lying (0). More.

Monotonic Concession Protocol (MCP)

There are the following 3 basic questions that should be answered with respect to the Monotonic Concession protocol:

- **What should an agent's first proposal be?**
- **On any given round, who should concede?**
- **If an agent concedes, then how much should it concede?**

What are the answers to these questions? Explain.

The negotiation in MCP consists of a series of rounds where each agent picks a deal from the possible deals and negotiates.

What should an agent's first proposal be?	It should be its most preferred deal so that it can make a concession later on.
On any given round, who should concede?	The agent least willing to risk conflict. At each round, agents calculate the utility lost by conceding and the utility lost by not conceding and the one who would lose the most should concedes.
If an agent concedes, then how much should it concede?	Just enough to balance the risk. The agent should make the minimum sufficient concession.

What is the basic idea of the Zeuthen strategy? Explain a formula for calculation of risks in this strategy.

Zeuthen strategy is used by the Monotonic Concession Protocol to understand who should concede.

$$\text{Risk}^{t}_{i} = \frac{\text{utility } i \text{ loses by conceding and accepting } j\text{'s offer}}{\text{utility } i \text{ loses by not conceding and causing conflict}}$$

What are the advantages and disadvantages of MCP and Zeuthen strategies?

Advantages:

- Simple
- Stable
- Reflects the way human negotiations work.
- Doesn’t guarantee success but guarantees termination
- Doesn’t guarantee to maximize social welfare but if an agreement is reached then it’s Pareto optimal

Disadvantages:

- Assumes that agents are rational and work on maximizing their utilities

Dominant strategies

What is a dominant strategy? Give an example of a dominant strategy.

In a dominant strategy an agent is best off by using a specific strategy no matter what strategies other agents use. For example, in the rock-paper-scissors game, where every possible choice has the same probability of being selected, the dominant strategy is to play randomly as the agent can do no better than playing randomly.

Nash equilibrium

Explain the notion of Nash equilibrium. Give an example.

In a Nash equilibrium, each agent chooses a strategy that is the best response to the other agents’ strategies. Two strategies s_1 and s_2 are in Nash equilibrium if:

- under the assumption that agent i plays s_1, agent j can do no better than play s_2
- under the assumption that agent j plays s_2, agent i can do no better than play s_1

Why is Nash equilibrium a good property of a negotiation protocol?

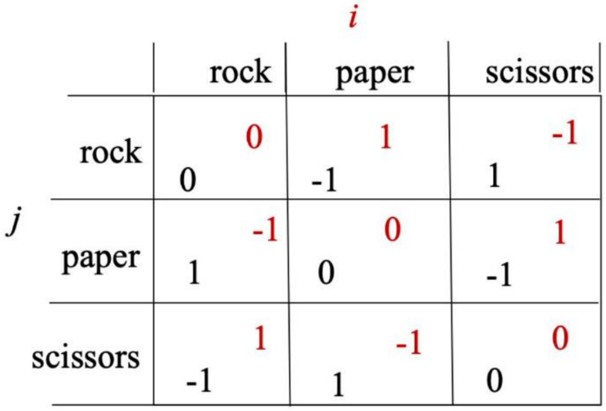
Because neither agent has any incentive to deviate from a Nash equilibrium, thus agents choose their strategies easily.

What is the difference between a dominant strategy and Nash equilibrium?

Use answers above. It is possible for a dominant strategy to be also nash equilibrium. if you want to explore the topic https://www.investopedia.com/ask/answers/071515/what-difference-between-dominant-strategy-solution-and-nash-equilibrium-solution.asp

Explain Nash equilibrium in the heads-tail coin game and the rock-paper-scissors game.

- **In the heads-tail game there is no Nash equilibrium because in each situation someone can do better than the other.**
- **In the rock-paper-scissors game there is no pure Nash equilibrium as there is no option in which both players' options are the best response to each other.**



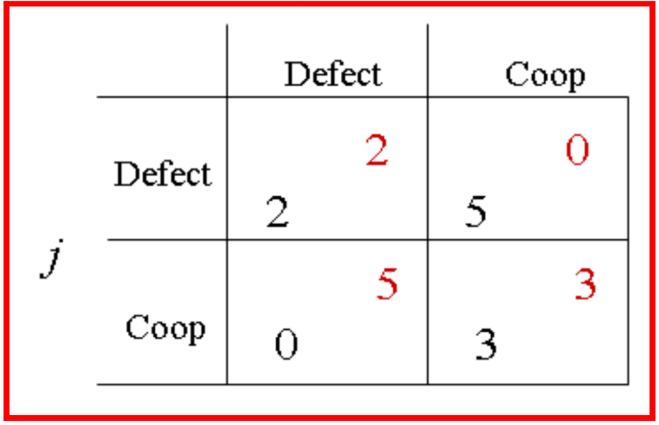
		i		
		rock	paper	scissors
	rock	0 0	1 -1	-1 1
j	paper	-1 1	0 0	1 -1
	scissors	1 -1	-1 1	0 0

- **However, by introducing randomness into the selection, with each choice having equal probability of being selected, we have a Nash equilibrium mixed strategy.**

What is Nash equilibrium in mixed strategy? Which property does it have?

- **Nash equilibrium in mixed strategy introduces randomness into the agents' selection of possible choices.**
- **Property: a strategy is in Nash equilibrium with itself, if each choice has the same probability of being selected, since it makes sense for everyone to play a random strategy in response to a random strategy.**

Draw a payoff matrix for the Prisoners dilemma (exact numbers should not necessarily be the ones presented in the lecture notes). Which cells in this matrix correspond to: social welfare, Pareto optimal solution and Nash equilibrium?



		Defect	Coop
	Defect	2 2	0 5
j	Coop	5 0	3 3

- Maximizing social welfare solution: CC
  - Why: it has the maximum utility (sum is 6)
- Pareto optimal: all but DD
  - Why: because one agent cannot increase its utility without the other agent decreasing its utility
- Nash equilibrium: DD
  - Why: because if agent i plays Defect, agent j cannot do better than playing Defect. If agent i plays Coop, agent j can do better by playing defect.

Why is the solution obtained by using Nash equilibrium strategy in the Prisoner's dilemma not completely satisfactory for participants?

Because cooperate/cooperate is not a scenario where agents maximize their own utility, but rather it's the best they can do given the strategy the other agent is playing.

Task-oriented and worth-oriented domains

What is the difference between task-oriented and worth-oriented domains?

- Task-oriented domains are domains in which agents negotiate how individual tasks should be split among each other, for example postmen that need to deliver letters to different locations and want to travel as little as possible.
- Worth-oriented domains are domains that require tasks to be carried out together and agents usually negotiate on multiple attributes, for example the time and location of a meeting they both need to attend.

What are worth-oriented domains?

Domains where agents assign a worth to each possible state + Use answer above.

What are the differences between hidden, phantom and decoy tasks?

These tasks can be used by agents to cheat in task-oriented domains. Let's consider the postmen domain, where each postman has to deliver letters and wants to travel as little as possible.

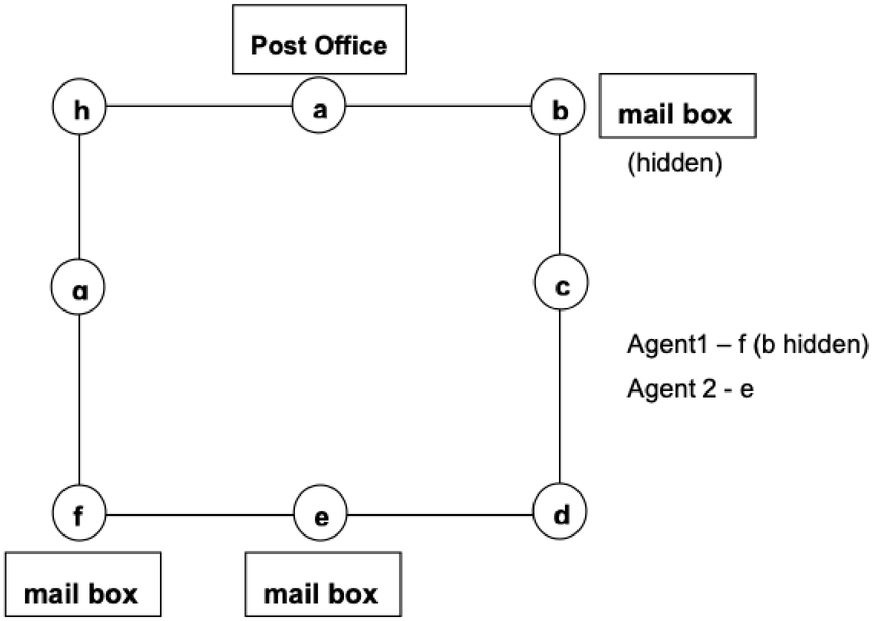
- Hidden tasks are letters agents hide to cheat and justify a shorter path.
- Phantom tasks are non-existent letters used by agents to cheat and justify a longer path. Since they don't exist, other agents may realize you're cheating. Cheaters can therefore make use of decoy tasks.
- Decoy tasks are letters generated on demand used by agents to cheat and justify a longer path.

What are some ways to get rid of deceptions in task-oriented domains? Explain how it will work for phantom, decoy and hidden tasks.

There are two ways:

- Mixed deals specify a probability distribution over partitions.
- All-or-nothing deals are mixed deals where one agent handles the tasks of all agents.

In class we considered an example of conventions for negotiation in the postmen domain. For example, see the figure below, where Agent1 should deliver letters from a to f and b and Agent2 should deliver a letter from a to e. However, Agent1 may hide the letter to be delivered to b.



We considered the usage of probabilities and all-or-nothing deal agreement to help get rid of deceptions. Demonstrate that with using probabilities and all-or-nothing deal an expected cost for Agent1 of performing its task with hiding letter is greater than expected cost of performing the task without deception.

Hiding letter:

- Expected cost = (3/8)*8 + (5/8)*2 = 4.25
  - Why?... expected cost is ivi . p(xi), where p_i is the probability of v_i and v_i is a value, in this example, we used **a way** of calculating p_i **which is dependent on the weight of the task(s) involved, so:**
  - We have 2 possible outcomes when hiding task b:
    - doing e, f, and b
    - doing b only
  - we used ⅜ as probability here, it is also the relative cost for the task f when hiding task b (3 steps out of 8 total possible steps).
  - 8 is the actual cost for doing tasks f, e and b, going all the way to e and back through b (8 total steps).
  - 5/8 is 1 - 3/8, as probabilities need to sum to 1.
  - 2 is the cost of doing the hidden task b only, one step to b and one back.
- Utility = 8 - 4.25 = 3.75

Non-hiding letter:

- Expected cost = (0.5)*8 = 4
- Utility = 4

The utility of not hiding the letter is higher, thus cheating by hiding letters is not convenient.

Contract Net Protocol

The basic premise of the CNP is that if an agent cannot solve an assigned problem using local resources, it will decompose the problem into subproblems and try to find other willing agents with the necessary expertise to solve these subproblems. Each agent can be both a manager and contractor.

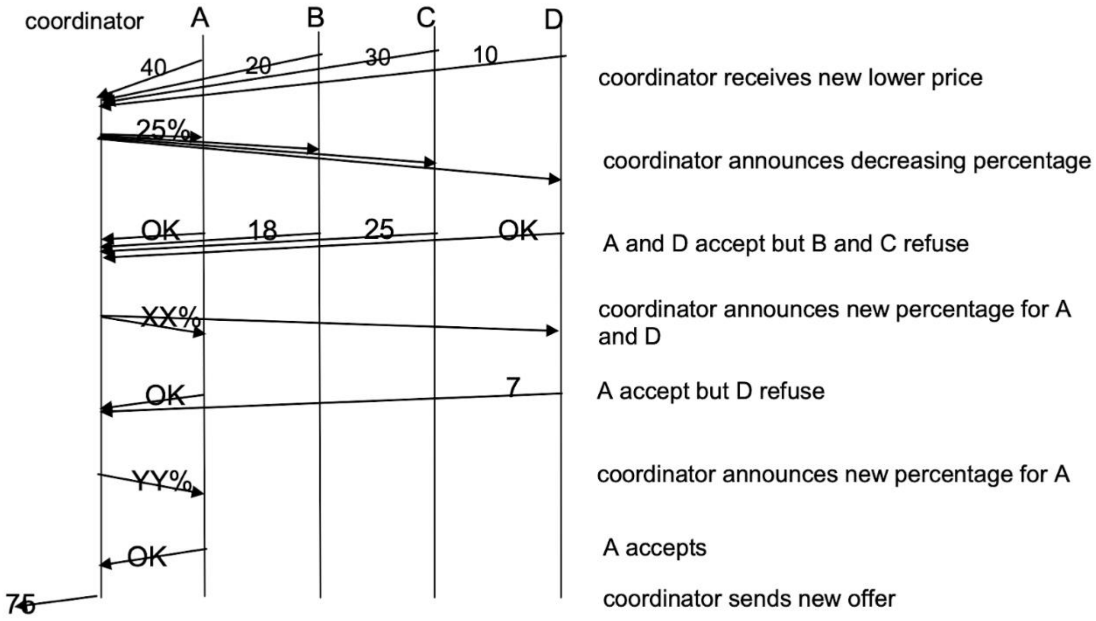
Give the advantages and disadvantages of the use of the Contract Net Protocol (CNP).

- **Simple.**
- **Decentralized.**
- **Doesn't detect conflicts.**
- **Assumes benevolent and non-antagonistic agents.**
- **Communication intensive.**

In class we considered an extended CNP for the case of a distributed system for resource managing in a building construction company (Foseca et al). One step in this example was intra-coalition negotiation according to the following algorithm:

1. **coordinator calculates the percentage of the initial cost that coalition must decrease**
2. **announcing percentage**
3. **agents respond by accept or reject**
4. **if there are rejects then new percentage calculation and goto 2**
5. **coordinator informs the announcer about new coalition position**

What will be the percentage for decreasing costs denoted by XX and YY on the figure below? Explain your calculations.


coordinator
A
B
C
D
40
20
30
10
coordinator receives new lower price
25%
coordinator announces decreasing percentage
OK
18
25
OK
A and D accept but B and C refuse
XX%
coordinator announces new percentage for A and D
OK
7
A accept but D refuse
YY%
coordinator announces new percentage for A
OK
A accepts
75
coordinator sends new offer

A, B, C, D send a new lower price to the coordinator

- **Initial value: 40 + 20 + 30 + 10 = 100**
- **New value decreased by 25% = 75**

A and D accept, but B and C refuse

- **B + C = 18 + 25 = 43**
- **A + D was initially 50, should now total 50 - 32 = 18, which is 36% of 50 (XX%)**

A accepts, D refuses

- **B + C + D = 18 + 25 + 7 = 50**
- **A was originally 40, should now total 75 - 50 = 25 and should be reduced by 40 - 25 = 15 which is 37.5% of 40 (YY%).**

2b. Auctions

An auction takes place between an **auctioneer** and a collection of **bidders**. The auctioneer wants to allocate goods to one of the bidders with the general objective to maximize the price. The bidder wants to buy goods that are as cheap as possible.

How can auctions be classified?

Value of goods:

- **Private (art)**
- **Public (stocks)**
- **Correlated (house)**

Winner determination:

- **First price**
- **Second price**

Bids may be:

- **Open cry (everyone can see bid)**
- **Sealed (only auctioneer can see bid)**

Bidding may be:

- **One shot**
- **Ascending**
- **Descending**

What is the best bidding strategy for each auction type that you know?

Similar question:

What is a bidder's dominant strategy in Vickrey auction? Prove your answer. What are the benefits of this strategy? What are is/her problems with Vickrey auction?

English auction (i.e. eBay, etc.)

- First price (you must pay the price you bid)
- Open cry (everyone knows the current price)
- Ascending (price can only increase)
- Susceptible to *winner's curse* (winner may have overvalued the goods)

- **Bid strategy: bid a little higher than highest current bid until it reaches true value, then withdraw**

Japanese auction

- Auctioneer starts with some price
- Each agent decides whether they're in or not by staying or leaving a room
- Agents can't go back once they step out of the area
- Auctioneer keeps increasing price until only one agent is inside area
- **Bid strategy: see what other bidders do**

Dutch auction

- Open cry (everyone knows the current price)
- Descending (price can only decrease)
- Auctioneer keeps lowering price until a buyer accepts current price
- Susceptible to *winner's curse* (winner may have overvalued the goods)
- **Bid strategy: bid a little below willingness to pay**

First-price sealed-bid auctions

- Single round, one shot auction
- Sealed: bids are shared to auctioneer privately
- Good is awarded to agent that made the highest bid
- If there multiple units for sale, bids are sorted from high to low and awarded to highest bidders until supplies last
- **Bid strategy: bid a little below true value**

Vickrey auctions (also called second-price sealed-deals)

- Second-price (the winner pays the price of the second highest bid)
- Sealed: bids are shared to auctioneer privately
- Good is awarded to agent that made the highest bid
- **Bid strategy: bid the true value because if you bid more you may end up paying too much for the good and if you bid less there is a smaller chance of winning**

Interrelated auctions:

This type of auction is well explained in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence* (link in Books section above).

- **Bid strategy: agents incorporate full lookahead so they take into consideration what they already won to inform new bids. This way the optimal allocation is reached, and agents don't bid their true per-item-costs**

Does it make sense to make the Vickrey auction open-cry?

Similar question:

We considered the second-price-sealed-bids or Vickrey auction. Does it make sense to have second-price open-bids-auction? Justify your answer.

Similar question:

Insincerity of the auctioneer (even if they don't disclose their bids to others) may be a problem in the Vickrey auction. Illustrate this by an example.

No. Let's imagine an open-cry Vickrey auction where everyone knows the other agents' bids. Another agent proposes 10 SEK for a book. You intentionally bid an absurdly high price (1 billion SEK) that is almost guaranteed to be the highest to get the book at 10 SEK (second price). Again, second-price means you pay the *second* highest proposed price, not your proposed price. However, the problem is that if there is more than one insincere auctioneer that bids high prices, you may end up paying a lot for the item.

What auction type(s) is/are most close to the basic CNP schema? Explain why.

It's the sealed-bid first-price auction:

- **announcing**
- **proposing bids**
- **evaluation**
- **awarding**

3. Agent Communication

Book references

- Chapter 7 of Wooldridge covers all of this quite well.
- Ontology is in Chapter 6 (optional).
- Bonus: paper that introduced KQML (covers routers & facilitators).
- Bonus: interoperation (around page 37)

Software interoperation

What is interoperation?

Interoperation is the ability of heterogeneous agents to communicate with each other to solve problems that they wouldn't otherwise be able to solve individually.

We considered 3 important aspects of a language: syntax, semantics, and pragmatics. Briefly explain them and give examples.

- **Syntax: how the symbols are structured in a language. It represents the common rules that must be followed to use any agent communication language. Example: in first order predicate calculus, the symbol must come before the terms: ()**
- **Semantics: what the symbols denote, their meaning. Example: the symbol ">" means the first term is greater than the second term.**
- **Pragmatics: how the symbols are interpreted. It's necessary to make sure the agents interpret the symbols in the same way. For example, "=>" should be interpreted as "implication" by all agents and some agents should not refer to it as "equal to".**

Which properties distinguish ACLs from other languages?

- **ACLs have semantic complexity.**
- **ACLs can handle prepositions, rules and actions.**
- **ACL messages describe a desired state in a declarative language, rather than a procedure or a method.**

Speech Acts

What are the main components of a speech act? Give examples.

1. **Performative verb** represents the agent’s intention (e.g. request, inform, inquire)
2. **Propositional content** (e.g. “the door is closed”)

Show by examples that the same propositional content used with different performative verbs may represent different speech acts.

Speech Act	Please close the door	The door is closed	Is the door closed?
Performative verb	**request**	**inform**	**inquire**
Propositional content	**the door is closed**	**the door is closed**	**the door is closed**

Which types of speech acts (illocutionary acts) do you know? Give examples

Illocutionary act	**Illocutionary point**
Commissives	**“I will”**
Expressives	**“Excuse me”**
Directives	**“Close the window”**
Declaratives	**“I name this door the Golden Gate”**
Assertives	**“It rains”**

Explain how we can describe the semantics of speech acts. Give a brief example.

Speech acts represent physical actions, and their semantic can be described via pre- and post- conditions.

Example of semantics of Request speech act: request(S, H,). The aim of the Request act will be for a speaker S to get a hearer H to perform some action . Below we define the Request act.

Preconditions

- **S BELIEVE (H CANDO)**. The speaker must believe that the hearer of the Request is able to perform the action.

- S BELIEVE (H BELIEVE (H CANDO)). The speaker must believe that the hearer also believes that it has the ability to perform the action.
- S BELIEVE (S WANT). The speaker must also believe it actually wants the Request to be performed.

Postconditions

- H BELIEVE (S BELIEVE (S WANT)). The hearer believes that the speaker believes that it wants some action to be performed.

KQML (Knowledge Query and Manipulation Language)

KQML is a **message-based language** for agent communication. Thus KQML defines a common format for messages. Each message has:

- a **performative** (the message intention)
- a number of **parameters** (attribute/value pairs, which may be thought of as instance variables).

Example A of a sender asking about the price of IBM stock:

ask-one is the *performative* (it means the sender will ask another agent a question where exactly one reply is needed)	(ask-one
the **:content** field specifies the message content	:content (PRICE IBM ?price)
the **:receiver** specifies the intended recipient of the message	:receiver stock-server
the **:language** field specifies the language in which the content is expressed. Notice how KQML is just an “outer" language, and that you need another language for the content (KIF, LPROLOG)	:language LPROLOG
the **:ontology** attribute defines the terminology used in the message	:ontology NYSE-TICKS
	)

Other examples of performatives are `tell`, `ask-if`, `subscribe`, `unsubscribe`, `reply`, `forward`, etc. Other examples:

Can KQML be used as an agent development language? Explain.

No, it is an agent communication language and only has features for agent communication.

Is it possible to write a KQML message where the content is also expressed in KQML? Justify if not, give an example if yes.

Yes, it's possible to write an KQML message where the content is also expressed in KQML. In fact, the content of an KQML message could be written in any language including KQML. Example of KQML message that contains another KQML message:

```
(forward
    :from Agent1
    :to Agent2
    :sender Agent1
    :receiver Agent3
    :language KQML
    :ontology kmql-ontology
    :content (tell
        :sender Agent1
        :receiver Agent2
        :language KIF
        :ontology Blocks-World
        :content (On (Block A) (Block B))))
```

Explanation: Agent1 wants Agent3 to forward a message to Agent2.

What are facilitators? How can the performative "recruit" be implemented using a facilitator? Explain your answer using a diagram.

Facilitators are a feature of KQML and they are agents like all others that offer additional services, such as forwarding, brokering, recruiting and content-based routing.

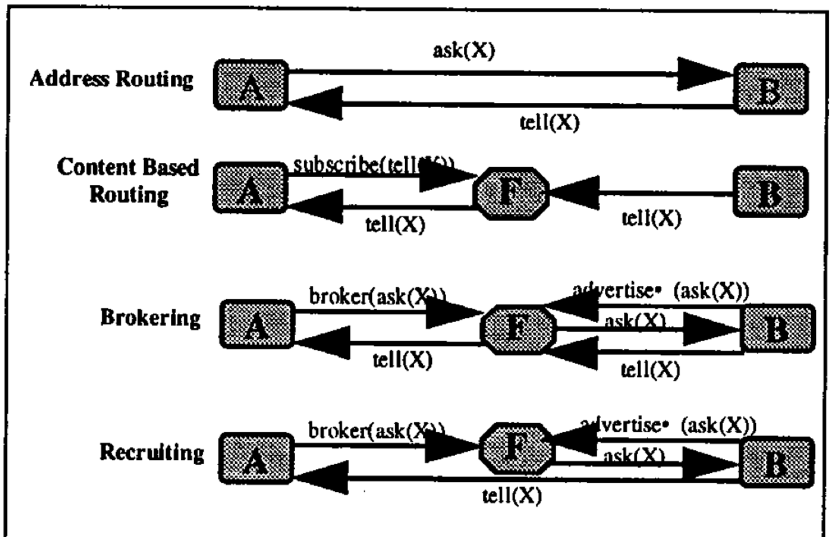


Figure 7 - Facilitators are agents that deal in knowledge about the information services and requirements of other agents. They can offer services such as forwarding, brokering, recruiting and content-based routing.

KIF (Knowledge Interchange Format)

What are the main motivations for developing KIF?
KIF is a language intended primarily to form the content parts of ACL messages. Its purpose is to facilitate the sharing of knowledge across heterogeneous systems that use different languages. It's not intended for human interaction, but humans can understand it.

What KIF looks like

```
(+ (sin theta) (cos theta))
```

Represents +.

```
(=> (believes John ?p) (believes Mary ?p))
```

Mary believes everything that John believes.

Example A of KQML using KIF:

```
(stream-about
    :sender A
  :receiver B
    :language KIF
    :ontology motors
    :reply-with q1
    :content (m1))
```

Explanation: Agent A sends a message to Agent B asking to stream everything about m1 (defined in the motors ontology).

Example B of KQML using KIF:

```
(ask-if
    :sender agent_i
    :receiver agent_j
    :language KIF
    :ontology genealogy
    :content “(spouse adam, eve)”
)
```

Explanation: agent_i asks agent_j if Eve is Adam's spouse. The genealogy ontology must be used to understand what spouse, adam and eve are.

FIPA: Foundation for Intelligent Physical Agents

As a result of KQML's pitfalls, the Foundation for Intelligent Physical Agents (FIPA) started developing another communication standard in 1995, called the **ACL language (Agent Communication Language)**, in competition with KQML.

Can FIPA ACL be used as an agent programming language? Explain.
No, it is an agent communication language and only has features for agent communication.

FIPA ACL message structure:

- **Envelope**: collection of parameters. Contains at least the mandatory `to` and `from` parameters, and may contain `intended-receiver` (if it's not defined, then the `to` parameter is copied over to `intended-receiver`)
- **Message body (ACL message)**

Example of a FIPA ACL message:

```
(inform
    :sender agentA
    :receiver agentB
    :content (price good200 150)
    :language KIF
    :ontology hlp-auction
)
```

Notice how this looks so similar to KQML. We can tell this is a FIPA message by looking at the performative (`inform`) which is only defined in FIPA. `inform` and `request` are the two basic performatives.

Is it possible to write an ACL message where the content is also expressed in ACL? If no, justify, if yes, give an example.

Yes, the content of an ACL message could be written in any language including ACL. Missing example: Prof said to look at slide 47 for the comm. slides pack.

FIPA ACL vs KQML

Give basic ideas of the agent communication languages KQML and FIPA ACL. What do they have in common and what are the basic differences between them?

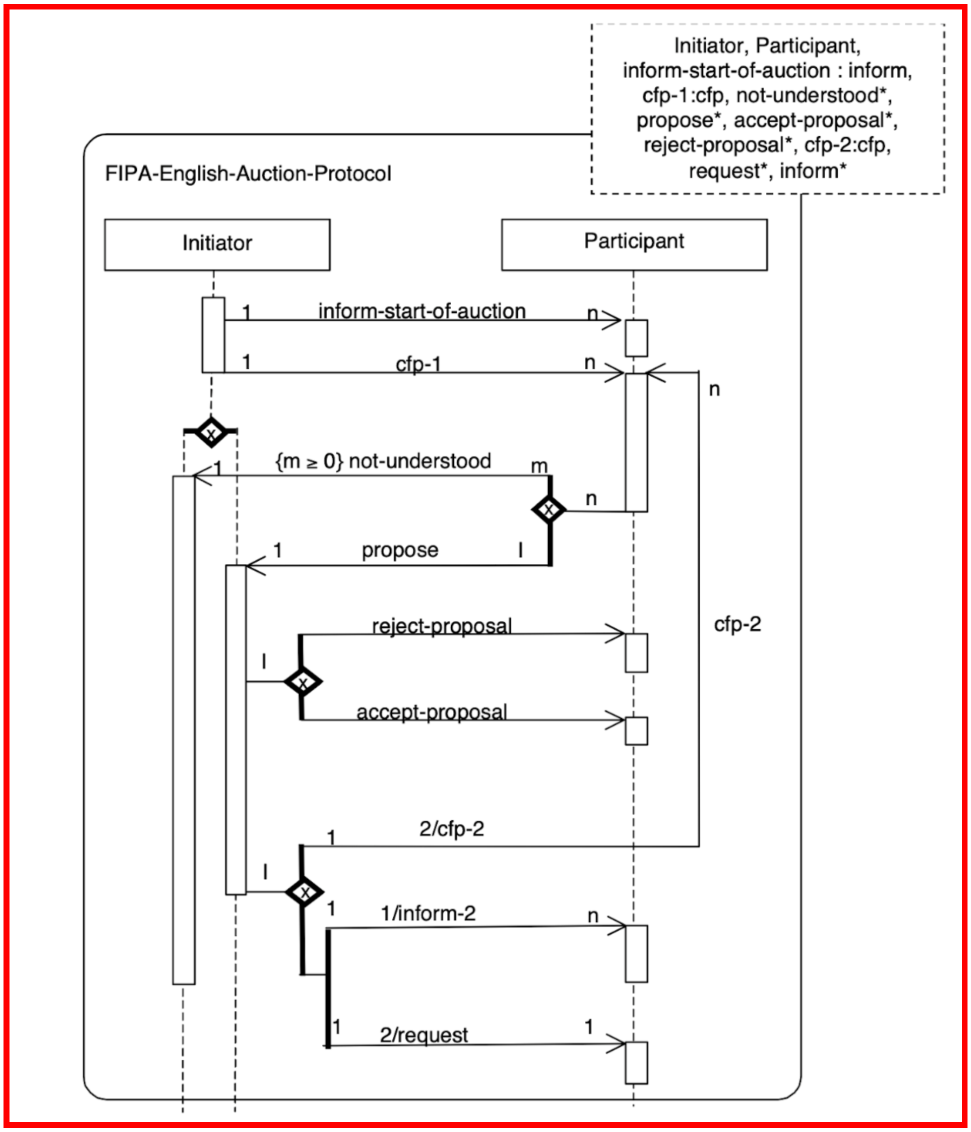
Similarities:

- **are both based on speech act theory.**
- **have a similar format (sender, receiver, content, language, ontology, etc).**
- **provide means for agents to communicate their intentions to other agents. In KQML, it's called performative, while in FIPA ACL it's called communicative act.**
- **are independent of the syntax and ontology used to describe the actual content of the message.**

Differences:

- **FIPA ACL includes a description of the pragmatics in both narrative form and formal semantics, whereas KQML doesn't give an explicit explanation on the performative.**
- **FIPA does not provide facilitator agents while KQML provides a class of agents that serve or facilitate other agents, such as a broker, translator or matchmaker.**
- **FIPA also introduced an Agent Interaction Protocol to provide clear guidance on how to use performatives and standardize common agent interactions.**

Describe the English Auction using FIPA ACL performatives.
Show the English Auction as an agent-interaction protocol diagram. Describe your answer.

- 

4. Agent Coordination

Book references

- Chapter 8 of Wooldridge.
- Other optional resources in the slides.
- Bonus: https://arxiv.org/pdf/1711.09057.pdf
- Bonus: A Survey of Self-organisation Mechanisms in Multi-Agent Systems
- Bonus: Cooperative Robots to Observe Moving Targets: A Review

Coordination Basics

Coordination is about following your own agenda while caring about others' agenda and behaviors. If the goal is to maximize your own utility only, then it's not a cooperative situation, otherwise if the goal is to maximize the joint utility then it's cooperated.

What are the fundamental coordination processes? Give a brief explanation.

1. **Mutual adjustment: each party adjusts its behavior taking into consideration the behavior of the partner.**
2. **Direct supervision: agents take turns to command (shared centralization).**
3. **Standardization: i.e. traffic rules.**

Give an example of a coordinated action that is cooperative.

Dancers that all have the goal of meeting in a central point when called by the choreographer.

Distinguish between coordinated action that is cooperative and not cooperative, using an example from driving in traffic.

- **Cooperative: cars drive in the same lane.**
- **Non cooperative: all cars try to avoid being an obstacle for an incoming ambulance.**

Is it always necessary that coordinated actions are cooperative? Give an example of a coordinated action that is not cooperative.

No, it is not necessary. A group of kids are playing in the park. As it starts to rain, all of them reach for shelter.

Does external observation of agent behavior always allow to recognize if the behavior is coordinated or not? Justify your answer by examples.

No, observing may not be enough to understand the behavior of other agents. Example: assume that there are two robots in one room with two doors. The first robot moves to the first door and the other assumes that it wants to leave the room, so it goes to the second door. In reality, the first robot just moved towards a waste bin near the door and after that it moves to the second door. Then the robots crash into the second door.

We considered coordination as an effective control of distributed search. What are the coordination activities in this case?

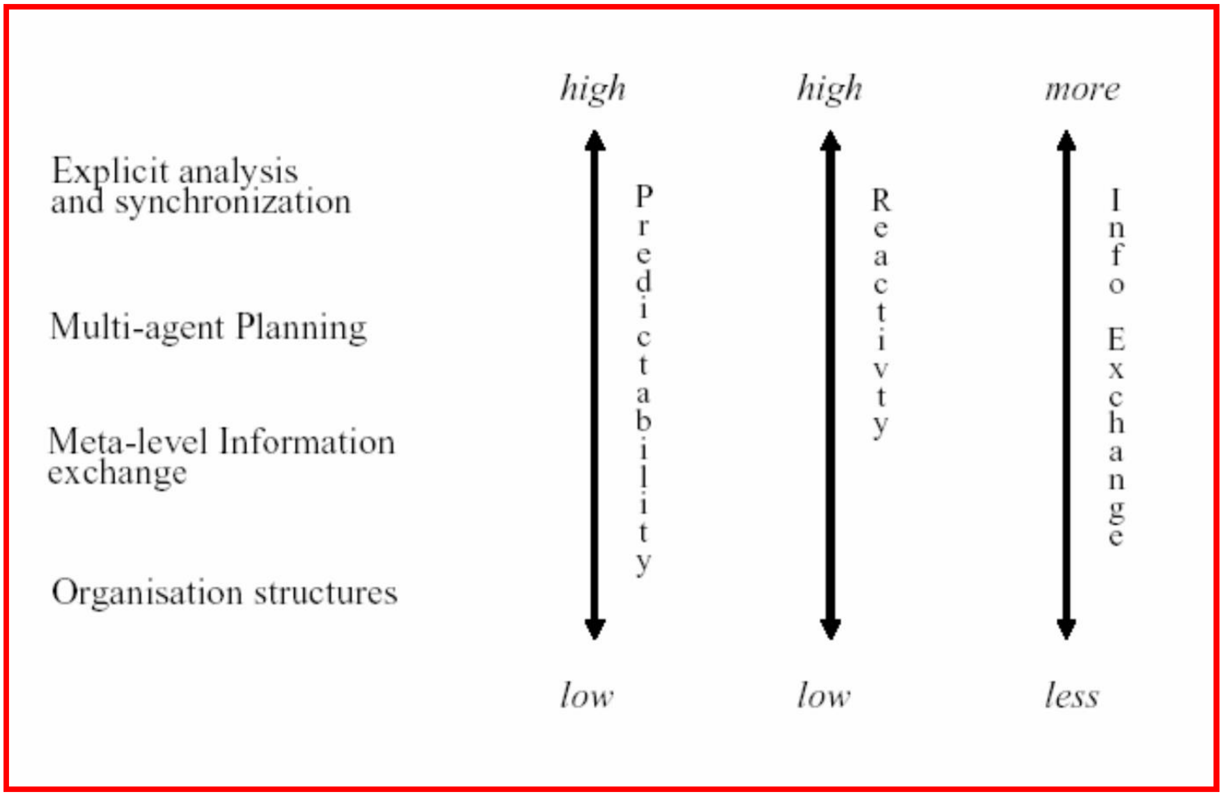
1. **Defining the goal graph**
2. **Assigning particular regions of the graph to appropriate agents**
3. **Controlling decisions about which areas of the graph to explore**
4. **Traversing the goal structure**
5. **Ensuring report of the successful traversal of the search space is reported**

Agents explore different parts of the search space, and coordination is needed if there are intersections between areas that agents need to explore (common goals). This scenario can be illustrated either by search space or search tree.

Coordination techniques

Name types of coordination techniques and range them according to predictability, reactivity, and information exchange. Compare common coordination techniques in terms of predictability, reactivity and amount of information exchange.

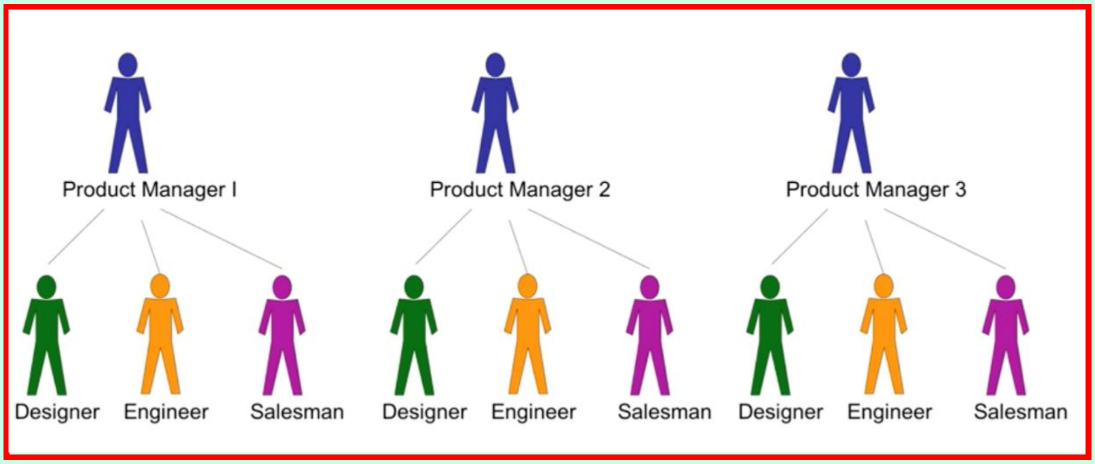
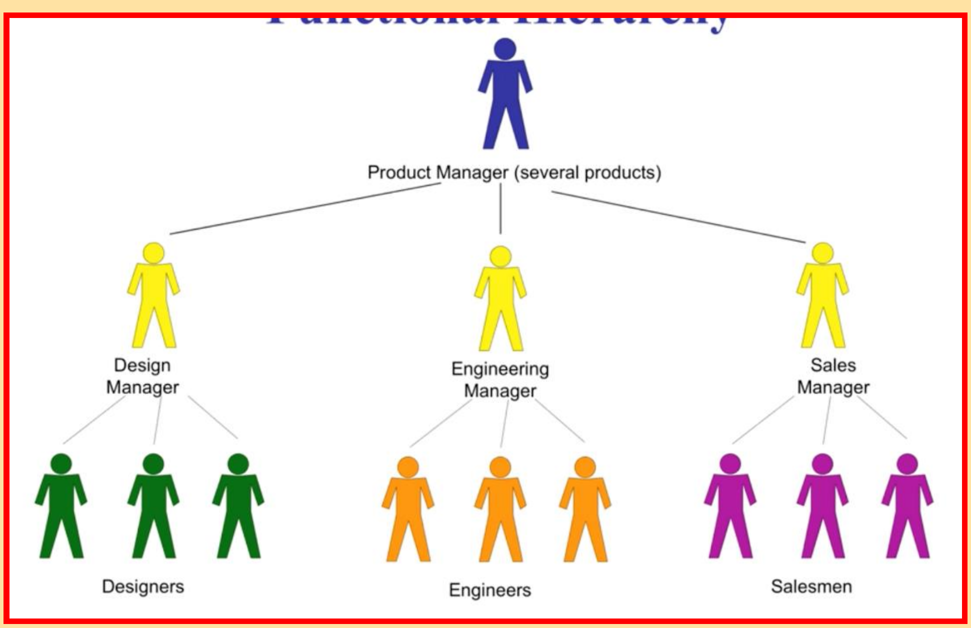
The most common coordination techniques are (see drawing below).

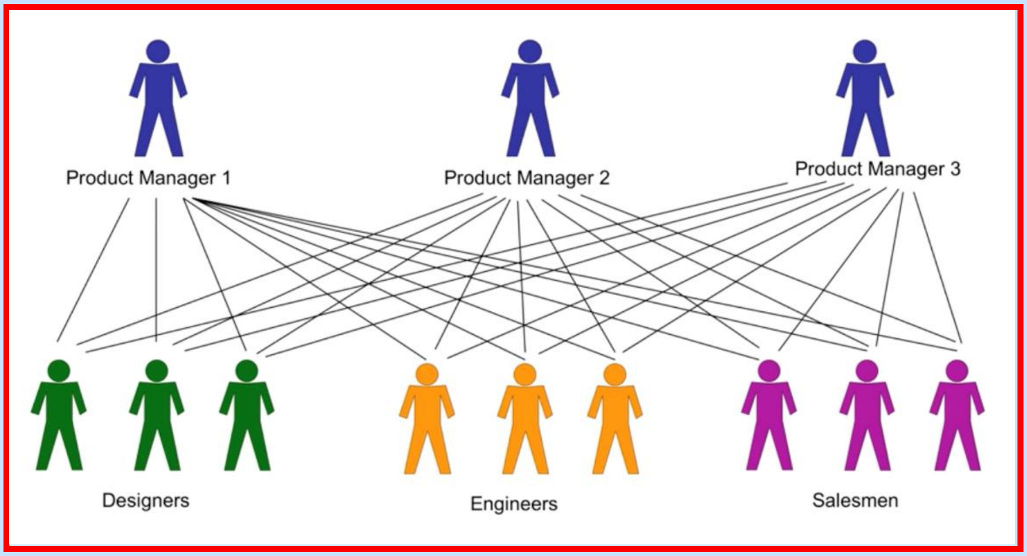
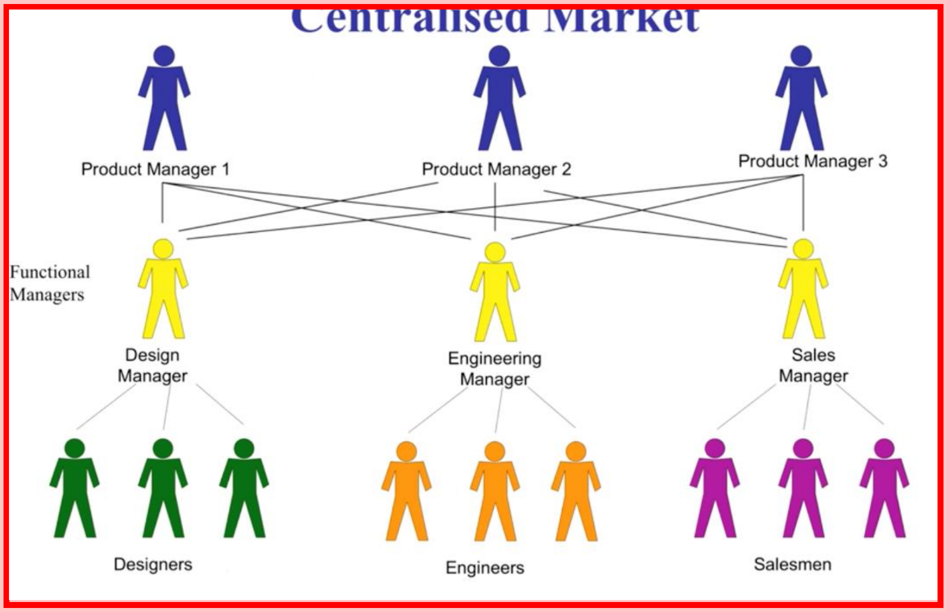


Organizational structures

Explain the main organizational structures that you know. Give their advantages and their disadvantages.

Product	Pros	Cons
Product hierarchy **PM for each team of ICs (Individual Contributors)**	**Low communication cost, failures don't affect other products as the teams are self-contained**	**The whole product is damaged if something happens to one of the agents, costly since all expertise is needed in each group**

<table>
<tr><td colspan="3">
</td></tr>
<tr><td><strong>Functional hierarchy PM at top, managers and ICs</strong></td><td><strong>Flexible: functional managers are good at assigning tasks, and replacing ICs if they don’t do their job well</strong></td><td><strong>The role of the manager becomes critical, especially the product manager, and is a bottleneck</strong></td></tr>
<tr><td colspan="3">
</td></tr>
<tr><td><strong>Decentralized market</strong></td><td><strong>Flexible: easy to choose and change the subcontractors</strong></td><td><strong>Very high communication cost, if the PM fails then the product fails</strong></td></tr>
</table>

 Product Manager 1 Product Manager 2 Product Manager 3 Designers Engineers Salesmen		
Centralized market	**Lower communication cost than decentralized market, managers choose the best supplier**	**Product is damaged if something happens to PM**
 Centralised Market Product Manager 1 Product Manager 2 Product Manager 3 Functional Managers Design Manager Engineering Manager Sales Manager Designers Engineers Salesmen		

Compare different organizational structures in terms of: 1) sharing processing agents 2) number of requires messages to be sent for task allocation and 3) results of failure of processing agents, functional managers and product managers.

	Processors shared among products	**Messages required to assign task**	**Result of task processor failure**	**Result of functional manager failure**	**Result of product manager failure**

Product hierarchy	No	2	1 product disrupted	-	1 product disrupted
Functional hierarchy	Yes	4	Task reassigned	All prod. disrupted	All products disrupted
Decentralized market	Yes	2m + 2 (m suppliers)	Task reassigned	-	1 product disrupted
Centralized market	Yes	4	Task reassigned	All prod. disrupted	1 product disrupted

Meta-level Information Exchange

Explain the characteristics of meta-level information exchange coordination? What are the main characteristics of meta-level information exchange coordination?
Agents exchange control-level information (e.g. "Hey, I'm working on goal X"), but not the work they are doing specifically. Meta-level information exchange includes partial global planning (PGP).

Let us consider Partial Global Planning. Why is planning partial? And why is this planning global?
In partial global planning cooperating agents exchange control-level information to reach common conclusions about the problem-solving process.

- **Planning is *partial* because the system does not (indeed cannot) generate a plan for the entire problem.**
- **Planning is *global* because agents form non-local plans by exchanging local plans.**

Each agent starts working inside each cell and coordination happens in the overlapping areas, which are used to understand how to *combine* local results.

Difference between organizational structures and PGP:
In organizational structures we assume there's someone who puts together the org structure and allocates agents. In PGP nobody has the general view, agents try to reach the global goal by using local communication.

Is it reasonable to use Partial Global Planning in air traffic control systems? Explain.
No, PGP is inappropriate for domains such as air-traffic control because it's dangerous to have gaps in coordination and PGP doesn't coordinate prior to execution.

Multi-agent planning

What are the main differences between Partial Global Planning (meta-level information exchange) and multi-agent planning?

- In PGP only control-level information is exchanged, in multi-agent planning complete plan of actions are exchanged.
- PGP does not require agents to reach mutual agreements before they start acting, multi-agent planning does.

Explain the following planning possibilities:

- **centralized planning of distributed plans**
- **distributed planning of centralized plans**
- **distributed planning of distributed plans**

Answer:

- centralized: a central coordinator develops, decomposes, and allocates plans to individual agents
- distributed: a group of agents cooperate to form:
  - centralized plan: each agent develops its own plan but they send it to a central coordinator who resolves conflicts (e.g. air traffic control)
  - distributed: both distribution and coordination happens in a distributed way

Explicit analysis and synchronization

Explain what is "explicit analysis and synchronization" in agent coordination. What are its pros and cons? In which situation it may be practically applicable?

Explicit analysis and synchronization consists of analysis of a situation in each decision-making step.

- Good: very flexible, and very useful if the level of dependency is low and the granularity of actions is high
- Bad: inefficient and has high communication costs during planning

Social norms vs laws

What are the differences between a norm and a social law in agent coordination?/Give examples of social norms and laws in agent coordination. What is the difference between them?

	Description	Example
Social Norms	Behavior one is expected to follow, but carries no authority	Standing in the queue while waiting for the bus
Social Laws	Behavior one must comply with, otherwise there will be consequences. They carry authority and can be seen as a set of constraints to agents	Driving on the right side of the road

5. Agent Theory

Book references:

- Chapter 17 of Wooldridge's book

Agents as intentional systems

We consider agents as intentional systems. When is such consideration most useful and when is it not appropriate?

Agents can be considered intentional systems when they portrait human attitudes, such as believing and wanting. It's not useful when we already know how the system operates (e.g. light switch).

Explain notions of "common knowledge" and "distributed knowledge". Give examples.

- **Common knowledge: we assume everyone knows this knowledge. Example: social laws.**
- **Distributed knowledge: nobody knows the whole knowledge, but everyone collectively does. Example: the 7 guardians of the Internet that can piece their keys together to form the password of the DNS.**

Modal logic

In modal logic we have two operators:

- □ **necessity**: if it's true in all accessible worlds
- ◊ **possibility:** if it's true in at least one accessible world

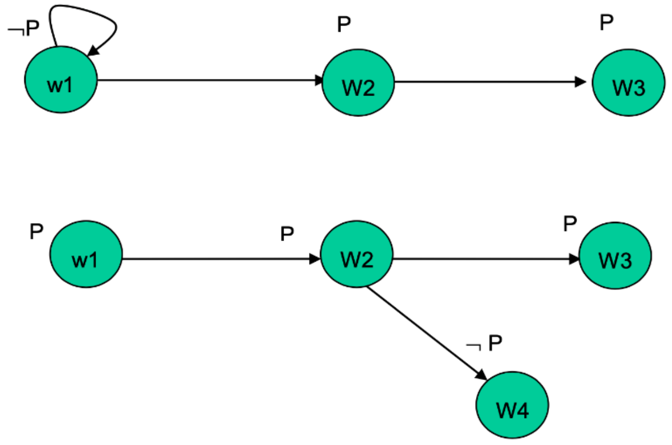
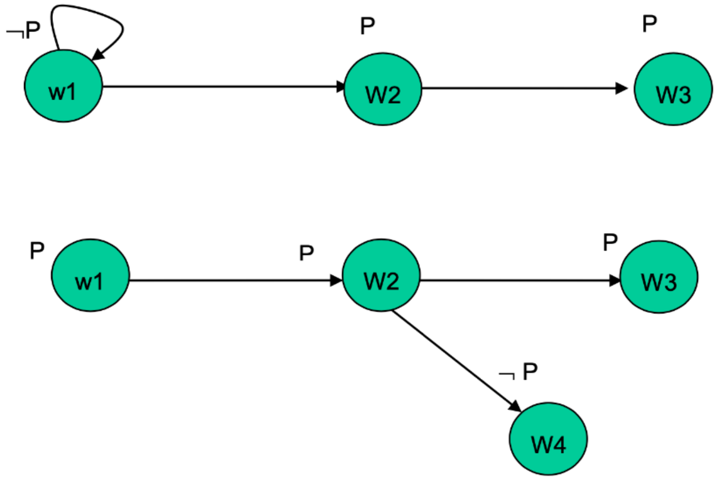
Are intentional notions truth functional? Explain your answer.

No, intentional notions are not truth functional: for example, believes(Mark, p) doesn't depend on whether p is true or false.

Is Modal Logic more suitable than Classical Logic for modelling intelligent agents? Explain your answer.

Yes. In classical logic the truthfulness of a formula depends on the truthfulness of its components, which is not suitable for intentional systems like agents. On the other hand, modal logic allows agents to express belief and knowledge through the necessity and possibility operators.

Express the following accessibility relations presented in the figures below using operators of modal logic:

 ¬P P P w1 W2 W3 P w1 P W2 P W3 ¬ P W4	**Transitive** w1: ◊P w2: □P **w1: ◊□P**
 ¬P P P w1 W2 W3 P w1 P W2 P W3 ¬ P W4	w1: □P w2: ◊P **w1: □◊P**

What are possible worlds? Give an example of a possible world.

- **Possible worlds are representations of the environment that the agent believes could exist. An agent may not be able to tell which one of a number of possible worlds describes the actual state of affairs. Something that is true in all possible worlds is knowledge.**
- **Example: three wise men below.**

In class we considered the Wise Man puzzle and the possible worlds. What are the possible worlds in this puzzle at the beginning and after each man answer?

There are three wise men.

- It is common knowledge -- known by everyone, and known to be known by everyone, etc. -- that there are three red hats and two white hats. The king puts a hat on each of the wise men, and ask them sequentially if they know the color of the hat on their head. Suppose the first man says he does not know; then the second say he does not know either.
- It follows that the third man must be able to say that he knows the color of his hat. Why is this, and what color has the third man's hat?

Number of red hats → 3
Number of white hats → 2
Number of men → 3

Possible worlds:

- **WHITE, WHITE, RED**
  - **The third man observes that the other two men wear a white hat, so he must necessarily wear a red hat since there only are two white hats.**
- **WHITE, RED, RED**
  - **The third man knows his hat is red because the second man isn’t able to figure out the color of his own hat, despite the first man wearing a white hat.**
- **RED, WHITE, RED**
  - **Similar reasoning to the above.**
- **RED, RED, RED**
  - **The first man can’t tell the color (so either the second or third person must be red), so doesn’t the second man (so either the first or third person must be red). This implies that the third men wears red.**

- **rww**
  - **Eliminated after first "don't know", if the first saw two other white hats, he would have said red.**
- **wrw**
  - **Eliminated after the second "don't know", if the second saw two other white hats, he would have said red.**
- **rrw**
  - **Eliminated after the second "don't know", if the second saw it the first wearing red and third wearing white, and given that /rww/ has been eliminated after first, then /r*w/ (* is either r or w) has only one possibility which is /rrw/, but this is not the case, so this is eliminated too.**

Logic of knowledge

Formalize in the logic of knowledge the following problem:

“Agent A wants to find out the cost of football tickets. Agent A doesn’t know the cost but Agent A knows that Agent B exists. Agent B doesn’t know the cost either but agent B knows that Agent C exists. Agent C knows the cost.”.

$(w_A p)(K_A p)(K_A E_B)(K_B p)(K_B E_C)(K_C p)$

where

- **w_A is agent A wants,**
- **p is the cost of the ticket,**
- **E_B is agent B exists,**
- **E_C is agent C exists.**

Formalize the following statements in the logic of knowledge (we consider the logic of knowledge as a modal logic with K_i operator).
Agent B knows that Agent C wants to find out the cost of football tickets:
$K_b w_c p$

If Agent A does not know the cost of football tickets then Agent B knows that Agent A does not know the cost:
$K_a p \rightarrow K_b(K_a p)$

Agent B doesn’t know whether Agent C knows that Agent B doesn’t know that Agent A wants to find out the cost of football tickets:
$K_b(K_c K_b w_a p)$ & $K_b(K_c K_b w_a p)$

Agent 1 knows that Agent 2 knows p, but Agent 2 doesn’t know that Agent 1 knows that Agent 2 knows p:
$K_1 K_2 p K_2 K_1 K_2 p$

Using the modal operator for belief B, formalize the following statement:
Per does not believe that Ole believes that it is not raining, but Per believes that he believes that Ole believes that is not Sunday.
...

How do you translate the following formula into the logic of knowledge? $\Diamond x \Diamond x$. (This is covered in lecture 4, around 13:00 and in review of exam)
whether x = $\Diamond x \Diamond x$
$\Diamond x = \Box x$
$\Diamond x = Kx$
Answer: $Kx(Kx)$

Let us assume that “whether x” is equal to $\Diamond x \wedge \Diamond \neg x$ in modal logic. Express in the logic of knowledge the following statements and explain your answer:
I don’t know whether I know the answer:
…

I don’t know whether I know the answer and I don’t know whether my neighbor knows the answer:
$\Diamond xx = Kx(Kx)$
$x_1 = K_1 x$ and $x_2 = K_2 x$
$K_1(K_1 x) K_1(K_1 x)$ K1…. look at slides or at exam recap

BDI Architecture

What are the relations between beliefs, goals, desires, and intentions in the BDI-architecture?

- Beliefs represent the environment characteristics, what the agents believe. Agents may obtain these beliefs from sensing their world
- Desires are goals, or some desired end states based on its beliefs
- Goals are the desires which an agent actively commits to achieving
- Intentions represent the current plan of action chosen

What is the relation between knowledge and belief in the works of agent theory?
Knowledge is true belief.

Discuss the appropriateness of the Modal logics axioms D, T, 4 and 5 for logics and knowledge and belief. What axiom(s) from logic of knowledge are not valid in the logic of belief? Explain..

- D axiom: VALID. The D axiom says that "If I know A, then I don't know A". Replacing know by believe still makes sense.
- T axiom (knowledge axiom): NOT VALID. T axiom says that "what is known is true". In logic of belief it would say "what is believed is true" which is wrong.
- 4 axiom (positive introspection): VALID. 4 axiom says "I know what I know" → I believe what I believe, thus it is valid.
- 5 axiom (negative introspection): VALID. 5 axiom says that "I don't know what I don't know" → "I don't believe what I don't believe", which also makes sense.
-

6. MAS Architectures

What are the main types of agent architectures? Give an example of each type.

- Abstract
  - Example:
- Deliberative
  - Example:
- Reactive
  - Example: thermostat
- Hybrid
  - Example:

Explain the advantages and disadvantages of symbolic, reactive, and hybrid agents.

- Symbolic:
  - very simple planning algorithms
  - answer to the question below
- Reactive: purely reactive agents, no memory, no reasoning or symbolic representation

  - works best in unknown environments
  - very simple behavior
  - difficult to predict
  - cannot plan ahead
  - assumes mutually exclusive rules and no rule conflicts
- Hybrid: combines both reasoning and reactive architecture
  - pragmatic solution
  - still no clear consensus, but a lot of similarity
  - no real methodology
  - no real theory

Originally, all agents designed within AI were symbolic reasoning agents. What are the key problems with them?

Two key problems:

- Transduction: translate the world into accurate symbols in time for that description to be useful
- Reasoning: how to get the agent to reason on this information in time to be useful

(from slide 23 Agent architectures)

Abstract Architectures

Describe input/output parameters of the following functions in the agent's abstract architecture: see, next, and action. Which of these functions will be used for specifying pure reactive agents and which of them will be used for specifying agents with state (symbolic)?

Similar question:

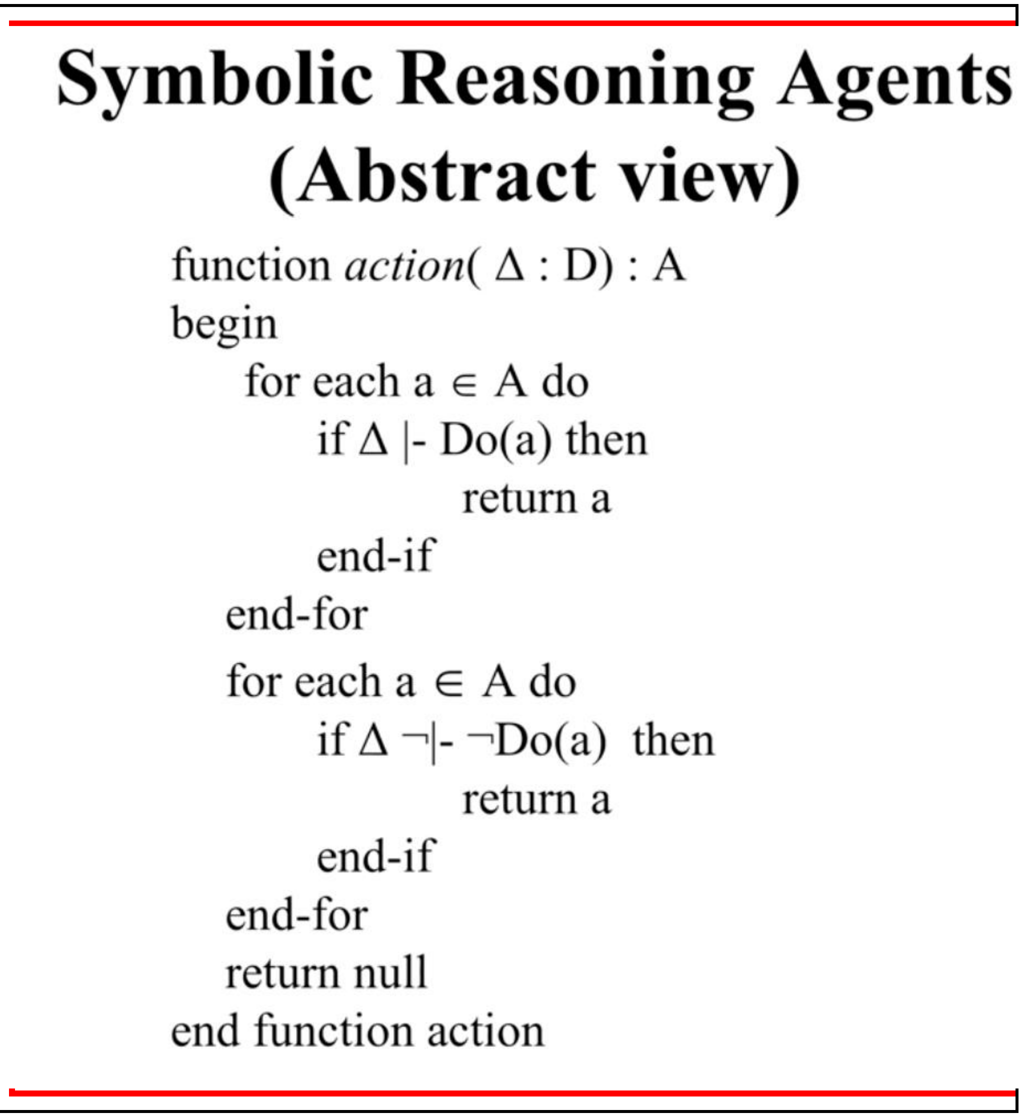
What is the meaning of introducing the "see" function in an abstract agent architecture? Give an example.

- see: agent's ability to observe its environment.
  - input: environment
  - output: percept
  - Used by both reactive and symbolic agents
  - Example: sprinkler agent "sees" fire in the environment and turns on the water.
- action: represents an agent's decision making process.
  - input: percept
  - output: action
  - Used by both reactive and symbolic agents
- next: stores the state
  - input: internal state and percept
  - output: internal state
  - Used by agents with state only (symbolic)

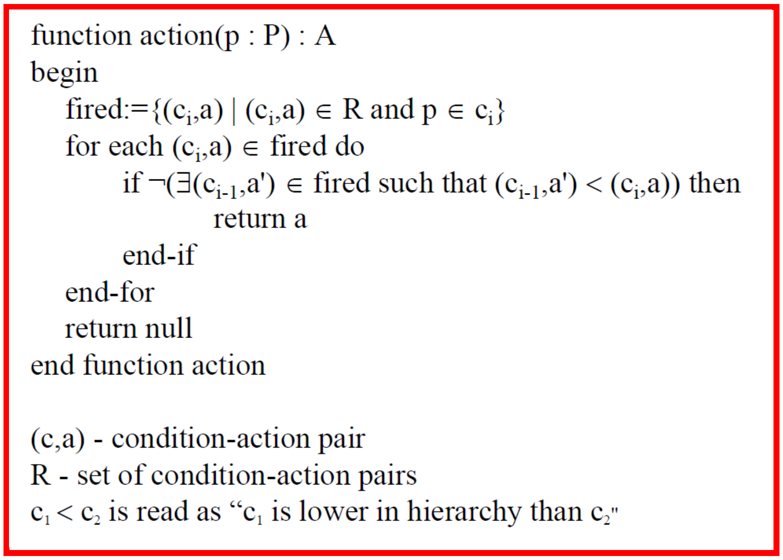
Give abstract implementation (in pseudo-code) of the function "action" for symbolic agents and for reactive agents?

Similar question:

Show an implementation (in pseudo-code) of action selection function for agents in the subsumption architecture?

Symbolic Reasoning Agents	 Symbolic Reasoning Agents (Abstract view)

```
function action( $\Delta$ : D) : A
begin
    for each a $\in$ A do
        if $\Delta$ |- Do(a) then
                return a
        end-if
    end-for
    for each a $\in$ A do
        if $\Delta$ ¬|- ¬Do(a)  then
                return a
        end-if
    end-for
    return null
end function action
```

Reactive agents (subsumption architecture)	

```
function action($p : P$) : $A$
begin
    fired:=$\{(c_i,a) \mid (c_i,a) \in R$ and $p \in c_i\}$
    for each $(c_i,a) \in$ fired do
        if $\neg(\exists(c_{i-1},a') \in$ fired such that $(c_{i-1},a') < (c_i,a))$ then
                return a
        end-if
    end-for
    return null
end function action
```

(c,a) - condition-action pair
R - set of condition-action pairs
$c_1 < c_2$ is read as “c_1 is lower in hierarchy than c_2"

Reasoning definitions

Explain concepts of practical reasoning, theoretical reasoning, deliberation and means-end analysis. Give examples.

- Theoretical reasoning using your knowledge and beliefs to acquire new knowledge and belief. Example: Socrates is a man, all men are mortal, therefore Socrates is mortal
- Practical reasoning is directed towards action, the process of figuring out what to do. Example: how to finish an MSc program

Human practical reasoning consists of:

- Deliberation: deciding what state of affairs we want to achieve. Example: when entering an MSc program you can decide the future career you want to have (industry or academia)
- Means-end reasoning: deciding how to achieve these states of affairs. Example: decide which courses to take to achieve the career

What is deliberation? Which phases it consists of and what is its output?

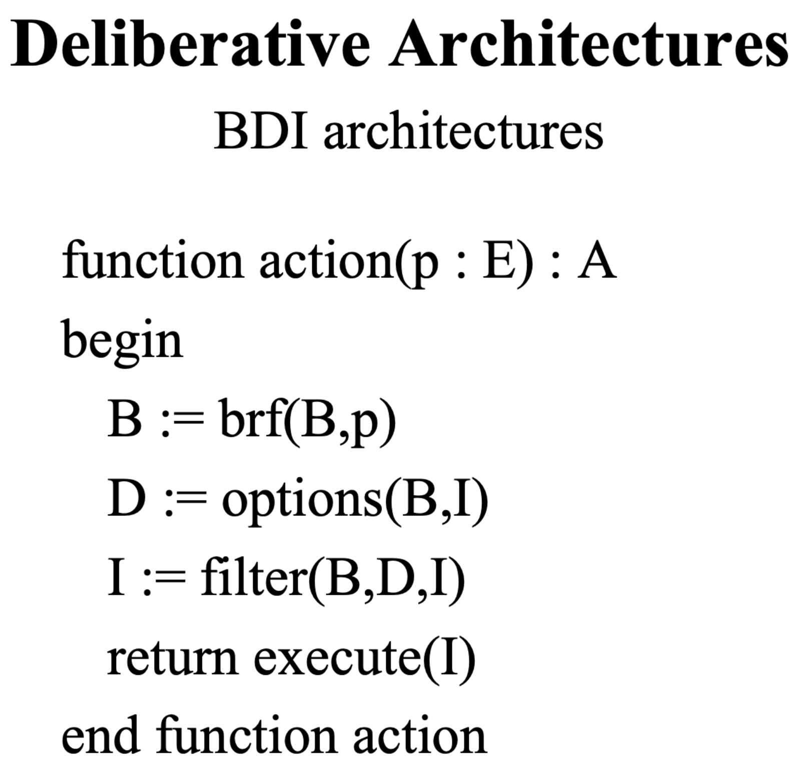
Deliberation is the act of deciding what goal to achieve, it consists of two phases:

1. Option generation: the agent generates a set of goals/desires
2. Filtering: choosing among the options and commit to it

The output are the chosen options which become intentions.

Deliberative Architectures

What would be an implementation (in pseudo-code) of the function “action” for an agent employing BDI architecture?



Deliberative Architectures

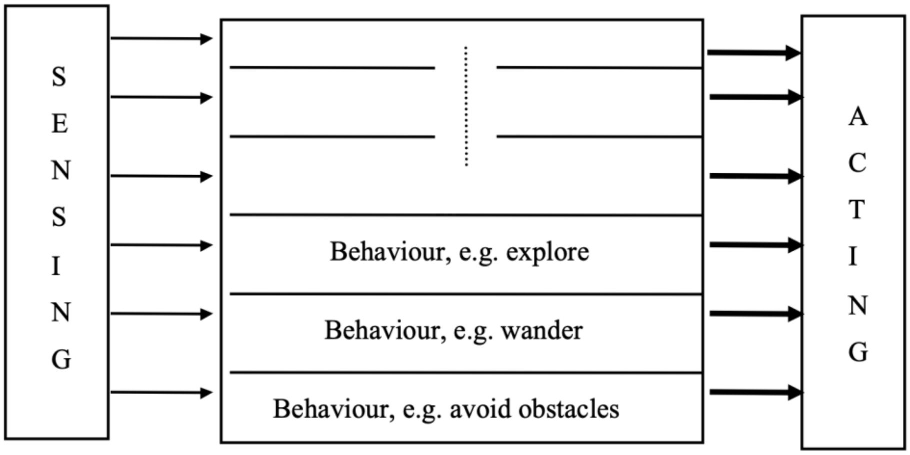
BDI architectures

```
function action(p : E) : A
begin
    B := brf(B,p)
    D := options(B,I)
    I := filter(B,D,I)
    return execute(I)
end function action
```

Reactive Architectures

Explain Brooks subsumption architecture. Give a sample example of its usage.

It’s a hierarchy of task-accomplishing behaviors. Each behavior, which looks like a rule, “competes” with others to exercise control over others.
This is opposed to the definition of AI with symbolic logic representation.



What kind of a layered architecture is Brooks subsumption architecture? Explain.

Horizontal - all layers are connected with inputs and outputs.

What is the difference between Brooks subsumption architecture and horizontal layered hybrid architecture?

The Brooks subsumption architecture

- doesn’t use any symbolic logic representation
- the lower “layers” have a priority over the layers on top.

What is/are the difference(s) between MAS architectures that use middle agents and market-based MAS architectures?

- Middle-agents: agents contact the middle agent and receive information which allows them to continue without the involvement of the middle agent (distributed).
- Market-based: centralized: agents communicate through a central market where negotiations are performed (centralized).

Hybrid agent architectures

What are the basic types of layered agent architectures? Explain their advantages and disadvantages.

- Horizontal: all layers are connected with input and output
  - Advantages: simple
  - Disadvantages: tricky to control
- Vertical: not all layers are connected to input and output. There are two types of architectures: one-pass and two-pass control architectures.
  - Advantages: control is pre-programmed
  - Disadvantages: less flexible, not fault-tolerant, difficult to implement because you need consistent interfaces

ACTOR Model

Explain how an ACTOR behaves.

Similar question:

Explain basic actions that ACTOR may perform.

An ACTOR is a message-driven system that sleeps until it receives communication. It has the following properties:

- Social: can send messages to other ACTORs
- Reactive: performs some behavior when it received a message from another ACTOR.

Basic actions it can perform:

- Sending messages to itself or other ACTORs
- Create other ACTORs
- Specify a replacement behavior, which is essentially another ACTOR that takes place of the ACTOR that creates it, for the purpose of responding to certain communications

Blackboard Architecture

What is the main idea of the Blackboard architecture? What are its main components?

The blackboard architecture is governed by two main ideas:

- Incremental: the solution to the problem is built step-by-step, agents build on top of other agents' input.
- Opportunistic: the system chooses the best action to progress

Main components:

- Blackboard: a global database of data and hypothesis
- Knowledge Sources: agents interacting with the blackboard
- Control mechanism: it decides the best possible action an agent should take and can be goal-directed and data-directed

Architecture design exercises

Propose a general architecture for a MAS which implements a simple virtual shop on the Internet. In particular, give the basic types of agents and possible types of coordination, communication and negotiation in such a system

The virtual shop has a number of buyers and sellers which connect to a shop and communicate through the shop.

- Multi-agent architecture: the shop is implemented by blackboard architecture which is goal directed. The agents post message about what they want/offer. The messages can be read and evaluated by all agents.
- Agent architecture: the buyer and seller agents can be implemented in hybrid architecture where the reactive layer is on the bottom and the deliberative layer is on top. Some decisions, while the other complex decisions, such as negotiation are made by deliberate layer.

Describe a layered, hybrid architecture that can support collaboration among agents as well as mobility of agents. Distinguish between the layers that are designed for the operation of the single agent and the ones that are designed for the collaboration within a multi-agent architecture. Explain with the help of a diagram.

…

7. Agent-Oriented Software Engineering

What is the rationale behind designing the AgentUML?

UML is not enough to model agents because:

- Agents are not objects, they are active
- Agents have complex interactions with other agents (negotiation, etc.)

Agents are active and act in cooperation and coordination with other agents, whereas objects just act in isolation.

What modifications to UML are proposed in AgentUML? Give examples.

Support for expressing concurrent threads of interactions, in order to model agent protocols as CNP.
Add the notion of role.

Show the English auction as agent-interaction protocol in Agent-UML.
Maybe same answer as the one in the communication section?

8. Mobile Agents

Give examples of situations (at least 2 situations) where mobile agents are appropriate (when they have benefits over communicating non-mobile agents)? Justify your answer.

Situation 1 (reducing Internet traffic)
You are deleting files on a remote server via SSH. Each thing you do generates traffic. If you instead send a mobile agent it is simply one message to the server and another one when the agent returns. The mobile agent would locally delete the files when arriving on another server. ← student's answer, needs check

Situation 2 (bad Internet connection)
You have a weak Internet connection which alternates between on and off every minute. You want to search for a file in a server. Instead of having to start over due to the Internet connection turning off, you can send a mobile agent which searches for it locally after arriving and then returns with the file when it is found and your Internet is up.

What is the difference between remote execution and a mobile agent?

- Remote execution is a program which is sent without execution state to a remote CPU, executes there, possibly communicates with other CPUs and then terminates.
- Mobile agent is a program which carries execution state with it and is sent to a remote CPU, executes there, possibly communicating with other CPUs, and then moves again to a third CPU or returns to its origin.

What are the main security issues for mobile agent systems?

- Authentication: both the agent and the host should be trusted.
- Secrecy: your agents should maintain your privacy.
- Security: the mobile agent can't be infected by virus or consume so much resource on the host.

Describe a simple way of implementing agent mobility using Remote Procedure Calls (RPC).

How could you implement agent's mobility if you can only use remote procedure call?