Tutorial 1:

Multi-agent Systems -- course and the research topic

- solving hard problems in an environment with other reasoning entities
- several derived topics (planning -> multi-agent planning; optimization -> decentralized optimization; …)
- several new and unique topics (game theory, auctions, voting, mechanism design)
    - these topics are closely related to economics and social sciences (study behavior of a group of people)
    - in our case -- agents
    - often (not exclusively) we are focused on the computational aspect
        - how to play in the game?
        - how to bet in an auction?
        - how to design an auction if we want to earn the most?
        - how to aggregate other agents' preferences to present to the user?

What is an agent? Connection between agents and robots -- a good way how to think about agents is to think about robots:

- robots are placed in some environment and react on the changes in the environment, need to deliberatively solve some task, need to communicate with other robots in the team, etc. …
- robots are currently available - you can easily purchase functional robots (vacuum cleaners, quadrotors, …) that you can deploy and use in practice

However, agents are more abstract. They, for example, can also be seen as an abstraction methodology -- consider a multi-agent simulation of a real-world large city and different level-of-detail of simulation based on the chosen autonomy/abstraction level.

Think about the robotic soccer -- how can you design a robotic soccer player?

- simple reactive agent -- (hierarchical) if-then rules
    - what happens? (miscoordination, deadlocks, …)
- coordinated agents
    - what happens? (better, but opponent takes the ball)
- reasoning about opponent
    - what happens? (fine, but intractable (the opponent will score while we are thinking what to do))
- decomposition of the team of cooperative agents

3 main topics/tasks that we will cover over the semester:

- coordination of **cooperative** agents
  - a group of agents must accomplish some common goal in an uncertain environment
  - formal model of reasoning -- modal logic, knowledge system, BDI
- solving problems in environment populated with **self-interested** agents
  - game-theory, solving the game
  - social choice, voting, auctions, mechanism design
- solving problems in cooperative setting, where agents wants distributively solve a common task
  - distributed constraint satisfaction programming
  - optimization

Rules for the tutorials

- Throughout the course we will use Java, MasSim, IBM CPLEX
  - Java -- simple, easy to debug, you should know it
  - MasSim -- simple environment that offer basic 'multi-agent robotic-like' environment
    - imperfect sensors
    - imperfect actuators
    - multiple agents
  - CPLEX
    - one of the best MILP/LP solvers
    - sure there are others, but this is not a competition :)
- Programming skills are required -- programming agents in a multi-agent environment involve problems from concurrent programming:
  - accessing shared memory
  - sending messages and their asynchronous parsing
  - outdated information about the environment
- Earning points for homework assignments
  - 9p + 14p + 17p = 40p all together, at least 20p for gaining the assessment
  - you will upload your solution via upload system
    - semi-automatic evaluation
  - deadlines are **strict!**

Tutorial 2:

Modal logic -- what is modal logic?

- informally, a logic which contains a modality, or several modalities
- a modal logic can be described with the notion of possible worlds
    - M = (W, R, V)
        - W - a set of possible worlds (states of the system)
        - R - accessibility relation (w, w' \in R, if world w' is reachable (or it is considered possible) from state w), we can denote this as (w R w')
        - V - evaluation of propositions in each world of W
    - basic modality: w⊨□p -> p is necessary true if and only if for every w', such that w R w', it holds that w'⊨p
        - this for example means that in every other world state that I consider possible, p must hold
    - we can pose some restrictions on the properties of relation R
        - reflexive
        - symmetric
        - transitive
        - serial
        - euclidean
    - different modal logics apply different restrictions on the relation R
        - e.g., in modal logic representing knowledge (relation R represents that agent i knows something - K_i) uses reflexive, symmetric, transitive and euclidean assumptions
        - temporal logics have serial restriction (there is always a next state)
    - different restrictions cause different axioms to hold
        - e.g., if there is no symmetric restriction, an agent can believe that *p* is true, although it could be the case that *p* does not hold in the particular state -- i.e., the agents beliefs can be incorrect
- an example with cards
    - we have 3 players (A,B,C), 8 cards = 4xA, 4x8
    - each player receives two cards and places them on her forehead without looking at them
    - each player is able to observe the cards of the other players
    - players take turns and they try to determine what cards they have on their foreheads
    - we assume all players are rational, they do not take guesses, and they do not lie
    - an example:
        - you are player C, you observe AA, 88; it is your turn (both players A and B have said that they did not know) -- can you determine what your cards are?
    - modeling the dynamic of this game using modal logic and possible worlds
        - the player C considers possible states (AA, 88, AA) - (AA, 88, 88) - (AA, 88, 8A)
        - without any other information the player C cannot determine what cards are on her forehead
        - when player A says that she does not know, player C can remove the state (AA, 88, 88) from the state of possible worlds -- if this would have been the case, the player A could have surely determined the cards on her forehead (she would have observed 88, 88 -- there is no other possibility than having AA), but A has

said “I do not know”, therefore this state is no longer possible.

- example with a robot -- Where is waldo?
  - we can use temporal logic to describe behavior/dynamics of an agent or a system, but also e.g., the goals
  - we need to specify the states, sensors (r_1 … r_4 -> in which region the robot is, s_w -> whether the robot senses waldo)
  - we can specify the goal as follows: $\Box \Diamond$ (r_2 OR s_w) AND $\Box \Diamond$ (r_4 OR s_w)
    - the robot has to look for Waldo repeatedly in regions 2 and 4 until it sees Waldo
    - $\Box \Diamond$ p - this means that p will hold infinitely often in temporal logics
  - we can use model checking tools to verify whether our robot based on some simple rules can actually accomplish the goal
  - alternatively, there are approaches that could generate behavior of agent/robot based on the specification of the environment and goals in (restricted variant of) temporal logic

Computational Social Choice

- summary of the voting protocols
    - scoring rules (plurality, Borda)
    - single transferable vote (STV)
    - pairwise elimination
    - plurality with runoff
- Condorcet winner
- Exercise
    - assume following preferences:
        - 3 agents: a > b > c
        - 2 agents: b > c > a
        - 2 agents: c > a > b
    - Which of the candidates is selected if we use plurality voting?
        - Borda?
        - Pairwise elimination with ordering
            - (a,b,c)?
            - (b,c,a)?
            - (c,a,b)?
    - Assume that we want to include a fourth candidate, “d”, into the preferences profiles. Can we do it in such a way “c” will be the winner under Borda voting rule?
- Definition of Condorcet loser
    - This candidate loses in pairwise comparison with every other candidate.
    - Let’s assume we are using plurality voting rule. Can the winner under plurality be the Condorcet loser?
        - If so, find an example of such a situation. If not, prove it.
    - What happens under Borda voting rule?
- Manipulation
    - strategic behavior in the voting setting
    - Classical problem -- If one agent knows the full preferences of other agents, how hard it is to calculate an insincere vote that can improve agent’s preferences.
    - Computational complexity is beneficial in this case.
    - P for simple voting rules and one manipulator (e.g., Borda)
    - NP for more complex rules (e.g., STV)
- Statistical Social Choice
    - we can use the social choice methods for aggregating opinions of other agents/people in order to find the “ground truth”
    - maximal-likelihood estimation principle -- we are seeking for a model (truth), for which it is the highest probability that the evidence (gathered votes) is as observed.
    - the simple voting rules (e.g., scoring rules) are correct estimators -- by aggregating the votes in this way we are guaranteed to find a correct model

Game Theory:

Consider a following two-player game: Player 1 founds a book. With a probability **p** it is a book about multi-agent systems, and with probability **(1-p)** it is a book about ice hockey. Player 1 sees what the book is about and decides whether she will give the book to the second player or not. If player 1 decides not to give the book to the player 2, the game ends, and both players receive zero utility values. In the other case (i.e., player 1 decides to give the book to the second player), player 1 wrap the book in the gift-wrapping paper and gives it to the player 2, who cannot see what kind of book it is. Player 2 can now decide if she accepts the gift or not. Player 2 wants to get the book about multi-agent systems, however, she does not want to refuse the book about hockey. Refusing the book about multi-agent systems, or accepting the book about hockey does not give any benefit to the second player. In general if the Player 2 accepts the gift, Player 1 will be happy, otherwise Player 1 will be sad since her gift was rejected.

Task 1:
Formalize this game as an extensive-form game -- draw the game tree, assign utility values from set {-1, 0, 1} to both players.

(hint: start with the chance node)

Task 2:
Transform the game to the normal form. Find all pure Nash equilibria. Do they depend on the exact values of **p**?