Intelligent Agents

Inference First Order Logic

Ute Schmid

Applied Computer Science, Bamberg University

last change: 9. Juli 2012

The Idea of MAS

- An agent is a computer system that is capable of independent action on behalf of its owner or user.
- A multiagent system consists of a number of agents which interact, typically by exchanging messages via some computer network infrastructure.
- Different agents might represent users/owners with different goals/motivations.
- Therefore, to succesfully interact, agents require the ability to
  - Cooperate
  - Coordinate
  - Negotiate

  with each other (similar to interaction of people in everyday live)

Key Research Questions

- Micro/Agent Design: how to build agents capable of independent, autonomous action
- Macro/Society Design: how to build agents capable of interacting with other agents, esp. if they have different goals/interests?

- Standard AI: focus on intelligent individual
- MAS: Social abilities
  - Emergence of cooperation in a society of self-interested agents
  - Language to communicate beliefs and aspirations
  - Conflict recognition and resolution
  - Coordination of activities to reach common goals

Example Scenarios

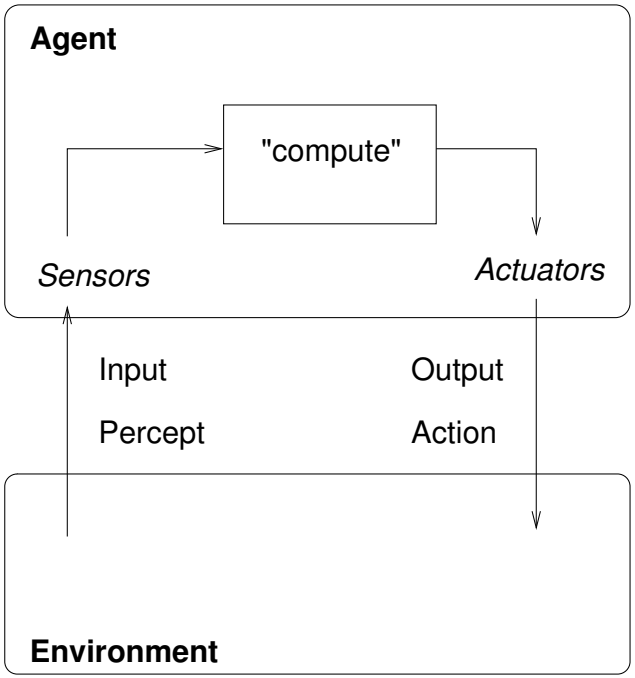
- NASA Deep Space 1 mission (1998): space probe with an **autonomous**, agent-based **control system** which can make some decisions by itself (before: control decisions were completley done by a 300 person ground crew)
- Autonomous air-traffic control systems: recognition of failure of other control systems and **cooperation** to track and deal with attended flights (e.g. DVMT, Durfee; OASIS)
- Last minute holiday package via PDA, using a **negotiating** agent

MAS is Interdiciplinary Research

- Software Engineering: Agent paradigm
  (going beyond OO)
- Social Sciences: Using theories, gaining insights by simulation of artificial societies
- AI: use planning, reasoning, learning technologies; study intelligent behavior in dynamic, interactive environments
- Game theory: use theories and techniques for negotiation

Definition: Agent

- An agent is a computer system that is situated in some **environment**, and that is capable of **autonomous action** in this environment in order to meet its design objectives.



Compare to

- Control systems: e.g. thermostate
- Software demons: e.g. xbiff

Environments

- Accessible vs. inaccesible

  obtaining complete, accurate, up-to-date information about the environments state
- Deterministic vs. non-deterministic

  each action has a single, guaranteed effect, no uncertainty about the result of an action; note: highly complex deterministic environments must be handled as non-deterministic
- Static vs. dynamic

  environment remains unchanged except by performance of the agent
- Discrete vs. continuous

  fixed, finite number of actions and percepts

↪ Open Env: inaccessible, non-deterministic, dynamic, continuous

Reactive Systems

- Two sources of complexity of MAS: characteristics of environment and nature of interaction between agent and environment
- Reactive system: maintainance of interaction with environment, must be studied and described on a behavioral level (not a functional level, i.e. in classical terms of pre- and postconditions)
- Example: reactive planning systems
- Local decisions have global consequences
- Example: printer controller
  Simple rule: first grant access to process p_1 and at some later time to process p_2 is unfair, because it might never grant access to p_2

Intelligent Agent

- An intelligent agent is a computer system with the ability to perform actions independently, autonomously, and flexible (on behalf of a user or an owner).
- Flexibility means: being
  - reactive
  - pro-active
  - social

Demands and Examples

- Performing a useful activity on behalf of humans or organizations (cleaning roboter)
- Coexist/interact with humans (cleaning roboter)
- Be aware of social rules and norms (transportation robot)
- Coordinate activities (team of cleaning robots)
- Cooperate or compete (RoboCup)
- Entertain or educate people (games, tutor systems)

Acting Reactively

- If the environment of a program is static (known in advance), the program cannot fail (Compile-Time, Runtime)
- In the real world, changes occur, information is incomplete (dynamic system).
- A reactive system continuously interacts with its environment and reacts in time to changes
- Example: Java-Listener, BUT: here reactions do NOT take into account the current state of the environment, they are determined in advance
- Reactive systems can be modelled relative straight-forward: e.g. as stimulus-response rules

Acting Pro-actively

- means: generate goals autonomously, try to reach goals
- not only event-driven behavior but: act on one's own initiative

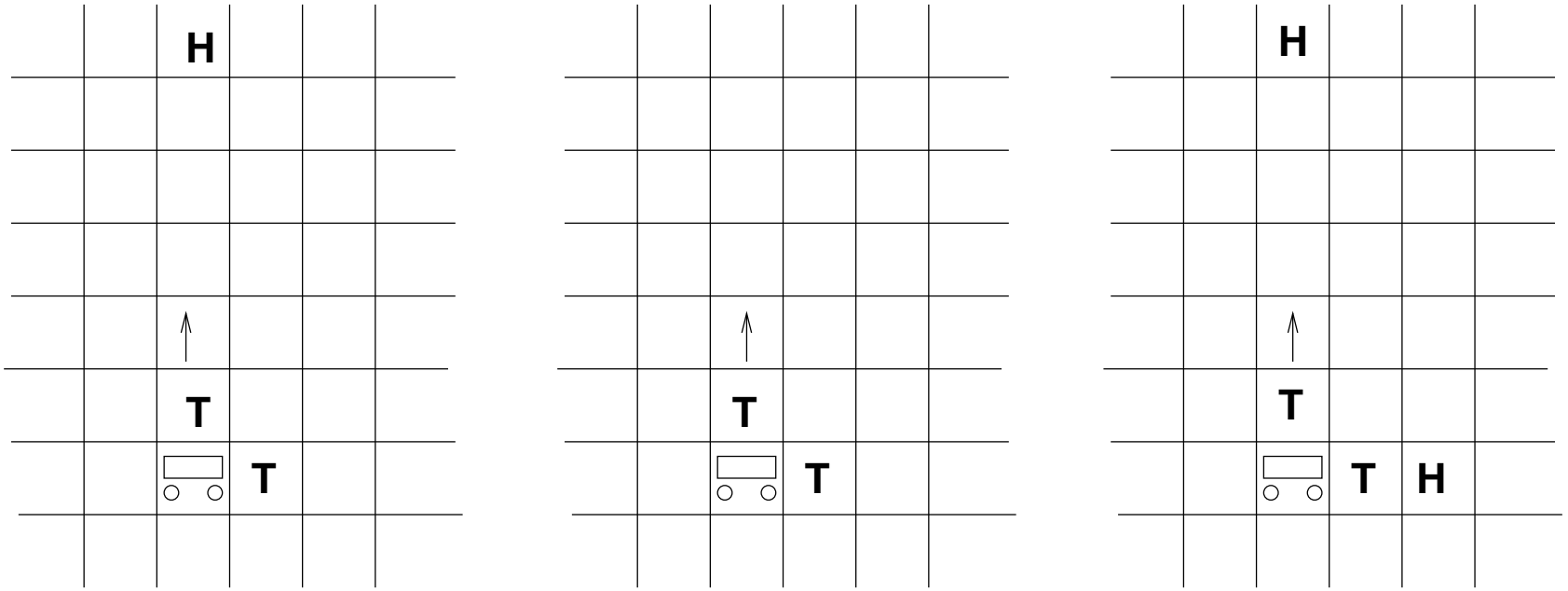
Acting Socially

- Real world is a multi-agent environment
- When trying to reach goals, others must be taken into account
- Some goals can only be reached through cooperation
- In some situations exit conflicts and competition (e.g. internet auctions)
- Social skills of agents: ability to model goals of other agents when trying to reach one's own (local) goals, ability to interact (i.e. cooperate and coordinate)

Further Features

- Mobility: ability to move in a computer net or in another environment
- Adaptivity/learning: Improving performance over time
- Rationality: do not act in a way which hinders to fulfill one's goals

Example: Tileworld



- Dynamic enviroment: holes appear/disappear
- Agent must recognize changes and modify its behavior

left: push up to H; middle: H disappears; right: better go to H to the right

Agents as Intentional Systems

- Endowing agents with “mental” states: beliefs, desires, wishes, hopes
- Folk psychology: attributing attitudes for predicting and explaining other peoples behavior
- Intentional systems (Dennett):
  - First order: having beliefs, desires, etc.
  - Second order: having beliefs and desires about beliefs and desires of its own and others
- Compare to physical systems: for predicting that a stone will fall from my hand I do nat attribute beliefs and desires but mass or weight

Abstract Architecture

- Environment: finite set of discrete states $E = \{e, e', \ldots\}$
  assumption: continuous env can be modelled by a discrete env to any degree of accuracy
- Repertoire of possible actions of an agent: $Ac = \{\alpha, \alpha', \ldots\}$
- Interaction of agent and environment: run r, as a sequence of interleaved environment states and actions
  - R: set of all possible finite sequences over E and Ac
  - R^{Ac}: subset of R ending with an action
  - R^{E}: subset of R ending with an environment state

Abstract Architecture

- State transformer function: $\tau : R^{Ac} \rightarrow P(E)$
- Termination: $\tau(r) = \emptyset$
- Environment $Env = (E, e_0, \tau)$
- Agent: $Ag : R^{E} \rightarrow Ac$
- Set of runs of an agent in an environment $R(Ag, Env)$
- Behavioral equivalence: $R(Ag_1, Env) = R(Ag_2, Env)$

Purely Reactive Agents

- No reference to their history, next state is only dependent on the current state
- $Ag : E \rightarrow \text{Ac}$
- Example: thermostate

Perception

- $see : E \rightarrow Per$
- $action : Per* \rightarrow Ac$
- Agent $Ag = (see, action)$
- Equivalence relation over environment states: $e \sim e'$ if $see(e) = see(e')$
- If $|\sim| = |E|$, the agent is omniscient
  If $|\sim| = 1$, the agent has no perceptual ability

Agents with State

- Internal states I
- $action : I \rightarrow Ac$
- $next : I \times Per \rightarrow I$

- State-based agents as defined here are not more powerful than agents as defined above.
- Identical expressive power: Every state-based agent can be transformed into a standard agent that is behaviorally equivalent.

Utility Functions

- Telling an agent *what* to do without telling it how to do it
- Indirectly via some performance measure
- Associate utility with states of environment, prefer actions leading to states with higher utilities
- Utility can be defined over states or over runs
  $u : E \rightarrow \mathbb{R}$ or $u : R \rightarrow \mathbb{R}$

Task Environments

- $\Psi : R \rightarrow \{0, 1\}$
  is 1 (true) if a run satisfies some specification and 0 (false) otherwise
- Task environment: $\langle Env, \Psi \rangle$
- specifies the properties of the environment and
  the criteria by which an agent will be judged to have succeeded in its task
- Definition of success
  - pessimistic: $\forall r \in R(Ag, Env)$ it has to hold $\Psi(r)$
  - optimistic $\exists r \in R(Ag, Env)$ where it holds $\Psi(r)$
- Two kinds of tasks:
  - Achievement: relation to planning
  - Maintainance

Deductive Reasoning Agents

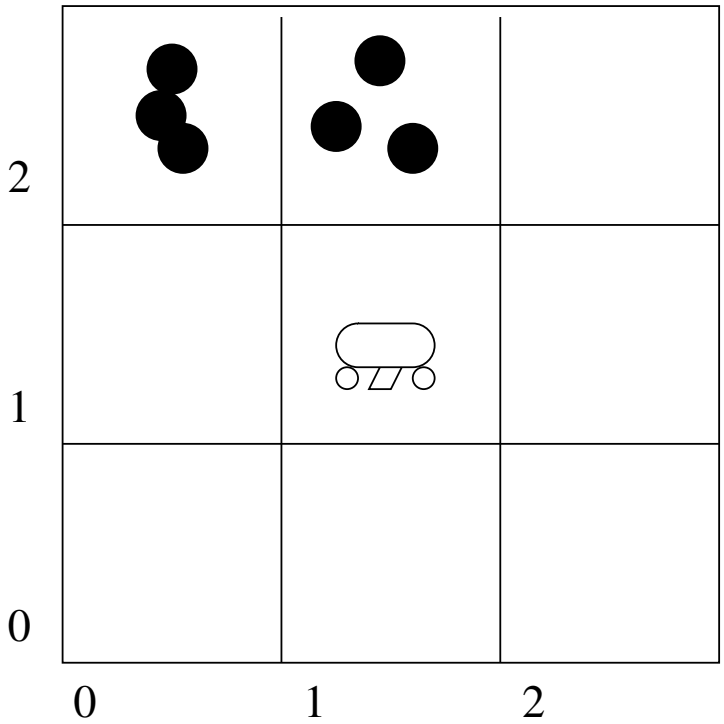
- The usual problems of knowledge engineering
  - Transduction problem: Translating world in adequate symbolic description
  - Representation problem: providing a representation such that agents can reason with it for the results to be in time and useful
- Agents as theorem provers
  Logic-based agents: “deliberate agents”
- Specialized languages, e.g. MetateM, based on temporal logic

Deliberate Agents

- D as a set of logical formulae, internal state of an agent is $\delta \in D$
- ρ as set of deduction rules
- $\delta \rightarrow_\rho \varphi$: formula φ can be proved from database δ using deduction rules ρ
- Goal: Deriving a formula $Do(\alpha)$ either as best action or as action which is not explicitly forbidden

- function action($\delta : D$)
- for each $\alpha \in Ac$ do
  if $\delta \rightarrow_\rho Do(\alpha)$ then return α
- for each $\alpha \in Ac$ do
  if $\delta \not\rightarrow_\rho \neg Do(\alpha)$ then return α
- return null

Vacuum World



Predicates:
In(x,y)
Dirt(x,y)
Facing(d)
(d $\in$ {north, east, south, west})

Navigation:
In(0,0) $\wedge$ Facing(north) $\wedge$ $\neg$Dirt(0,0) $\rightarrow$ Do(forward)

Cleaning:
In(x,y) $\wedge$ Dirt(x,y) $\rightarrow$ Do(suck)

Concurrent MetateM

- Michael Fisher, 1994
- Language for direct execution of logical formulae (exacutable temporal logic)
- Near the “ideal” of agents as deductive theorem provers
- Concurrently executing agents, communication via asynchronous broadcast message passing
- Components of an agent
  - Interface: defines interaction with other agents with *id*, set of *environment propositions* (accepted messages), set of *component properties* (messages the agent can send)
    `stack(pop, push)[popped, full]`
  - Computational engine (executable temporal logic)

Executable Temporal Logic

- Agent specification as set of program rules of the form `antecedent about past => consequent about present and future`
- Declarative past and imperative future paradigm

Temporal Connectives

Operator	Meaning
$\bigcirc\varphi$	φ is true 'tomorrow'
$\odot\varphi$	φ was true 'yesterday'
$\Diamond\varphi$	at some time in the future, φ
$\Box\varphi$	always in the future, φ
$\Diamond\bullet\varphi$	at some time in the past, φ
$\Box\bullet\varphi$	always in the past, φ
$\varphi U\Psi$	φ will be true until Ψ
$\varphi S\Psi$	φ has been true since Ψ
$\varphi W\Psi$	φ is true unless Ψ
$\varphi Z\Psi$	φ is true since Ψ
start	nullary operator, true only at the beginning

Example

rp(ask1,ask2)[give1,give2]:
$\odot$ask1 $\Rightarrow$ $\Diamond$give1;
$\odot$ask2 $\Rightarrow$ $\Diamond$give2;
start $\rightarrow$ $\Box\neg$(give1 $\wedge$ give2).
rc1(give1)[ask1]:
start $\rightarrow$ ask1;
$\odot$ask1 $\rightarrow$ ask1.
rc2(ask1,give2)[ask2]:
$\odot$(ask1 $\wedge\neg$ ask2) $\rightarrow$ ask2.

Example cont.

Example Run:

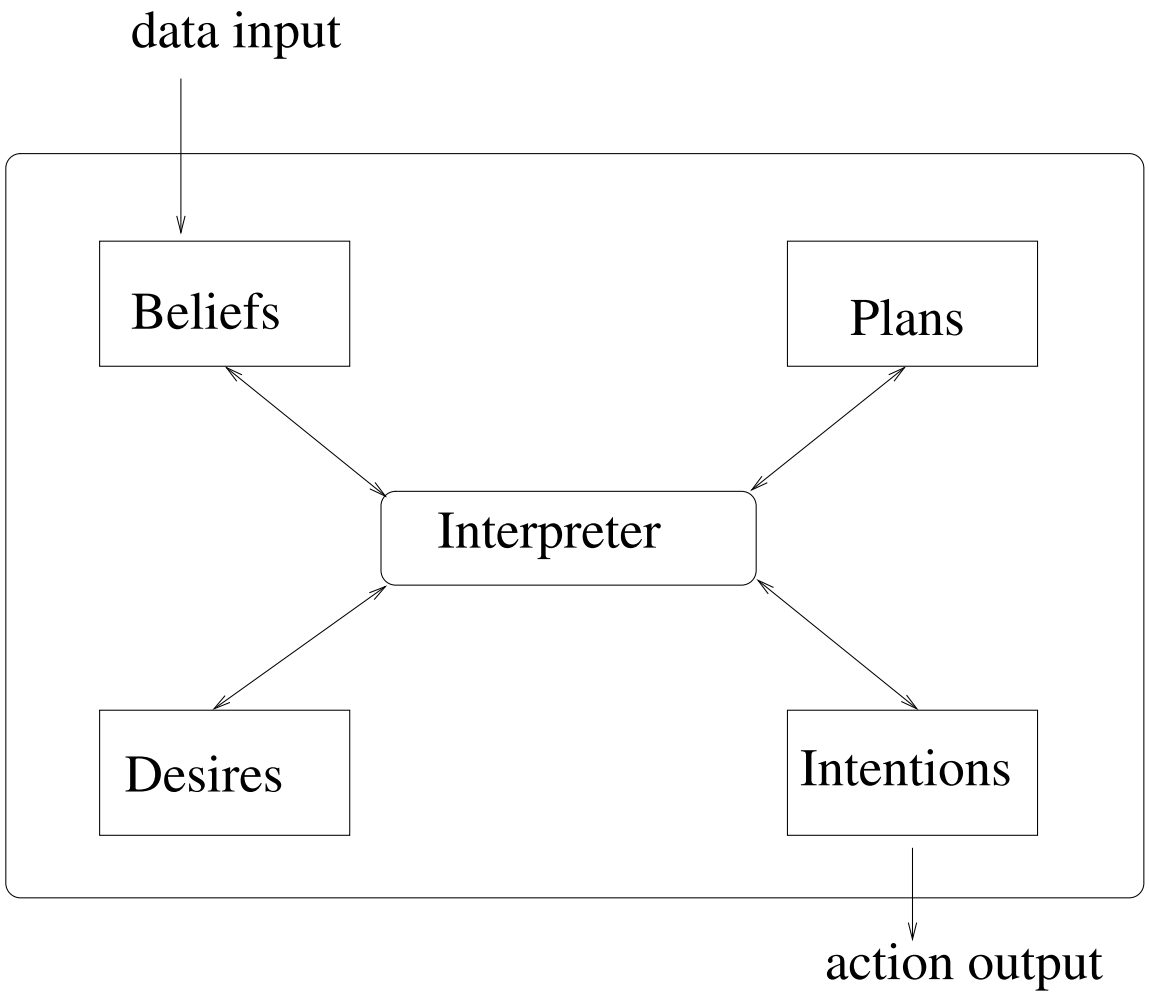
Time	Agent		
	rp	rc1	rc2
0		ask1	
1	ask1	ask1	ask2
2	ask1,ask2,give1	ask1	
3	ask1, give2	ask1, give1	ask2
4	ask1, ask2, give1	ask1	give 2
...			

Practical Reasoning Agents

- What we want to achieve: deliberation
- How to achieve a state: means-end-analysis

The Procedural Reasoning System

- Georgeff and Lansky
- Belief-desire-intention architecture (BDI)



Rational Actions

- Reasoning and planning agents: calculative/computational rationality
- First step: deliberation – what goal should be achieved forming an intention
- Second step: Means-End-Analysis – how should the goal be achieved
- Intention: agent commits itself to a task, inconstistent intentions can be blocked by an active intention
- Agents believe that their intentions can be fulfilled and do not belief that they cannot fulfill their intentions (rationality)
- Model: Belief, desire, intention Semantics (BDI)

In the following: Focus on multi-agent interactions (games)

Rational Actions in MAS

- Assume that environment can be influenced by both agents
- Utility function $u(\omega)$ with ω represented as action tuple:

$$u_i(D,D) = 1; u_i(D,C) = 1; u_i(C,D) = 4; u_i(C,C) = 4$$

$$u_j(D,D) = 1; u_j(D,C) = 4; u_j(C,D) = 1; u_j(C,C) = 4$$

- Preference of agent i:

$$(C,C) \geq_i (C,D) \geq_i (D,C) \geq_i (D,D)$$

- $\mathcal{C}$ is the rational decision for i, because it prefers all states where it selects $\mathcal{C}$
- This is the basic model of game theory.
- Representation in payoff matrices

Payoff Matrix

		i Coopearate	Defect
j	Cooperate	4 4	1 4
	Defect	4 1	1 1

Dominant Strategies

- Strategy: Decision function by which an agent selects an action
- A strategy s_1 dominates another strategy s_2 if an agent prefers each possible outcome of strategy s_1 over each possible outcome of strategy s_2 (with respect to the possible actions of the other agents)

		i	
		s1	s2
j	s1'	4 4	1 4
	s2'	4 1	1 1

outcome of $s_1 \geq$ outcome of s_2 for s'_1 and for s'_2

Dominance and Optimality

- A rational agent will never select a strategy which is dominated by another strategy (such strategies can be excluded when actions are selected)
- Unfortunately, there exists not always a distinct dominant strategy
- Pareto-optimality: an outcome is Pareto-optimal, if there is no other outcome all agents would prefer

Nash-Equilibrium

- Two strategies s_1 and s_2 are in a Nash-equilibirum if:
  - Under assumption that agent i plays s_1, s_2 is the rational choice of agent j
  - Under assumption that agent j plays s_2, s_1 is the rational choice of agent i
- If we have two strategies in Nash-equilibrium, no agent has an incentive to change its strategy
- Obtaining a Nash-equilibrium is desirable, because it is an effort to switch strategies and strategy switches might danger stability of a system

Analysis of game theoretic scenarios

- Nash-equilibrium is an important tool for analyzing game theoretic scenarios
- If each player has a dominant strategy, the combination of those strategies is called dominant strategy equilibrium
- Solution of a game: the rational strategy of each agent
- Each dominant strategy equilibrium is a Nash equilibrium
- Nash's theorem: there are equilibrium strategies even if there there is no dominant strategy
- Nash equilibrium is a necessary condition for an optimal solution (but is it also sufficient?)

Coordination Games

- Acme, a video game hardware manufacturer, must decide whether its next game machine will use CD or DVD
- Best, a video software producer, must decide whether to produce its next game on CD or DVD

		Acme DVD	CD
Best	DVD	9 9	-4 -1
	CD	-3 -1	5 5

Example cont.

- No dominant strategy
- Two Nash equilibria: (CD, CD) and (DVD, DVD)
  if one player moves to a different strategy unilaterally, he will be worse off
- One Pareto-optimal solution: (DVD, DVD)
- If payoff for CD and DVD would be equal, there would be two Pareto-optimal solutions

↪ guess or communicate
↪ coordination game

Zero-sum games

- If preference orders of two agents are diametral, we have a stronly competitive scenario:

$$\omega \geq_i \omega' \rightarrow \omega' \geq_j \omega$$

- An interaction is called zero-sum interaction, if

$$u_i(\omega) + u_j(\omega) = 0 \text{ for all } \omega \in \Omega$$

- All zero-sum interactions are competitive!

Prisoner's Dilemma

- Invented by Albert W. Tucker, further studied e.g. by Robert Axelrod
- Two people are accused of complicity in a criminal act
- They are in two different prison cells and cannot communicate
- The attorney guarantees:
  - If one confesses the crime and the other not, the first will be free, the other goes 5 years to prison
  - If both confess, both go for 3 years to prison
- Both know that they go 2 years to prison if none of them confesses

Prisoner's Dilemma

		i confess	i deny
j	confess	3 3	5 0
	deny	0 5	2 2

- Global utility is maximal if both cooperate (deny)
- But: for each single agent the rational choice is not to cooperate but to testify
- Is cooperation feasible in a society of rational, egoistical agents?

Generalized Form of PD

		i cooperate	defect
j	cooperate	win win	win much lose much
	defect	lose much win much	lose lose

- T: Temptation to defect, R: Reward for mutual cooperation,
- P: Punishment for mutual defection, S: Sucker's payoff
- For PD scenarios it must hold: $T > R > P > S$
- For iterated versions, in addition: $2R > T + S$
  If that condition does not hold, then full cooperation is not necessarily Pareto optimal, as the players are collectively better off by having each player alternate between cooperate and defect. (Douglas Hofstadter)

Iterative PD

- Repeated playing, memory of earlier encounters
- Studied in social psychology, competition of computed strategies
- greedy strategies tend to do very poorly in the long run, altruistic strategies do better
- Best deterministic strategy: “Tit for Tat”
  developped by Anatol Rapoport: the simplest of any program entered, containing only four lines of BASIC, won the contest!
- cooperate on the first iteration of the game; after that, the player does what his opponent did on the previous move

Iterative PD cont.

- Best if mis-communication is introduced: “Tit for Tat with forgiveness”
  When the opponent defects, on the next move, the player sometimes cooperates anyway, with a small probability (around 1%-5%). This allows for occasional recovery from getting trapped in a cycle of defections.

Analyzing Top Scoring Strategies

- Nice, retaliating, forgiving, non-envious
- Utopian sounding: Nice guys can finish first!
- e.g.: arms race

Dilemma of the Commons

- “Allmende Klemme” (William Forster Lloyd, 1833; Garrett Hardin, 1968)
- social trap that involve a conflict over resources between individual interests and the common good
- free access and unrestricted demand for a finite resource ultimately dooms the resource through over-exploitation!
- This occurs because the benefits of exploitation accrue to individuals, each of which is motivated to maximise his or her own use of the resource, while the costs of exploitation are distributed between all those to whom the resource is available
- Solutions?

Multiagent Communication

- Competitive: mechanisms for collective decision making
  - Voting
  - Auction
  - Negotiation
  - Argumentation
- Cooperative: communication for distributed problem solving
  - Speech acts
  - Agent Communication Languages
  - Ontologies

Collective Decision Mechanisms

Design of a protocol

- Guaranteed success: ensuring that eventually an agreement can be reached
- Maximizing social welfare: total sum of utilities should be maximal
- Pareto efficiency: no other outcome where at least one agent is better off and none is worse off
- Individual rationality: following the protocols in the best interest of negotiation participants
- Stability: providing all agents with an incentive to behave in a particular way (e.g. Nash equilibrium)
- Simplicity: a participant can easily determine the optimal strategy
- Distribution: designed such that there is no 'single point of failure' and to minimize communication between agents

Auctions

- Online auctions are very popular
- simple interaction scenarios $\hookrightarrow$ easy to automate
- good choice as a simple way for agents to reach agreements, allocating goods, tasks, resources
- Auctioneer agent, bidder agents, a good
- private value vs. public/common value of goods,
- correlated value: value for private factors as well as other agents valuation

Dimensions of Auction Protocols

- Winner determination: first price (highest bid gets good for the bidded amount), second price (highest bidder gets the good, but for price of 2nd highest bid)
- Knowledge of bids: open cry, sealed-bid
- Bidding mechanism: one shot, ascending/descending bids in successive rounds

$\hookrightarrow$ different types of auctions

English Auctions

- “Mother of auctions” (Sothebys)
- first-price, open cry, ascending (starting with a reservation price)
- Dominant strategy: successively bid a small amout more than the current highest bid until price reaches current valuation, then withdraw
- Winner’s curse: uncertainty about the true value (e.g. land speculation), winner might have overvalued the good

Dutch Auctions

- open-cry, descending
- auctioneer starts with an artificially high value
- decreasing value, until someone makes an offer
- no dominant strategy
- also susceptible to winner's curse

First-Price Sealed-Bid Auctions

- first-price, sealed bid, one-shot
- simplest form of auction
- difference between second-highest and highest bid is wasted money
- best strategy: bid less than true valuation, how much less depends on the other agents, no general solution

Vickery Auctions

- second-price, sealed bid, one-shot
- dominant strategy: bid true valuation
- Because truth telling is the dominant strategy, this form is discussed much in multiagent literature
- BUT: counterintuitive for human bidders
- Possibility of antisocial behavior: own valuation 90 $, guess that another agent will pay 100 $, therefore bid 99 $ such that opponent needs to pay more than necessary
- Commercial situations: one company cannot compete directly but foces other company into bankruptcy

Issues for Auctions

- Expected revenue: Strategies for the auctioneer to maximize his revenue (for risk-neutral bidders, expected revenue is provably identical in all auction types)
- Lies and collusions: coalition between agents (bid small amounts, share win afterwards), place bogus bidders, ...
- Counterspeculation: costs time and money and is risky (compare with meta-level reasoning)

Negotiaton

- Auctions are only concerned with allocation of goods
- When agents must reach agreements on matters of mutual interest, richer techniques are required
- Negotiation techniques for artificial agents (Rosenschein and Zlotkin, 1994)

Object vs. Agent Communication

- Object o_2 invokes a public method m_1 of object o_1 passing argument *arg*
  $\hookrightarrow$ o_2 communicates *arg* to o_1
  $\hookrightarrow$ BUT: the decision to execute m_1 lies only with o_2
- An autonomous agent has control over its state and its behavior
  There is no guarantee that another agent really performs an action
- An agent cannot force another agent to perform some action or to change its internal state
- An agent can try to influence another agent by communication
- Communication can change the internal state (belief, desire, intention) of another agent
- Communication as special case of action: speech acts