General

- 6 credit units
- Can be included in post-graduate studies

- Lectures (4h per week), 8 + 6½ weeks
- Student presentations instead of lectures towards the end
- Weekly exercises (2h per week)
- Programming exercise
- An essay + voluntary presentation

- You may have a chance to influence the contents now

- Lectures based on a new edition of the course book, not significantly different from the previous one
Organization & timetable

**Lectures:** prof. Tapio Elomaa
- Tue 12–14 TB219 & Thu 12–14 TB223
- Jan. 10 – Apr. 26
  - Period break: Mar. 5 – 11
  - Easter break: Apr. 5 – 11
  - Traveling?: Apr. 18 – 20
- Student presentations after Vappu (in May)

**Weekly exercises** starting on week 4 (Jan. 23 →)
- M.Sc. Teemu Heinimäki
  - Day, Room?

Exam: Fri May 18, 2012 1 – 4 PM

---

Topics

**Part I Artificial Intelligence**
- 1 Introduction
- 2 Intelligent Agents

**Part II Problem Solving**
- 3 Solving Problems by Searching
- 4 Beyond Classical Search
- 5 Adversarial Search
- 6 Constraint Satisfaction Problems

**Part III Knowledge and Reasoning**
- 7 Logical Agents
- 8 First-Order Logic
- 9 Inference in First-Order Logic
- 10 Classical Planning
- 11 Planning and Acting in the Real World
- 12 Knowledge Representation

**Part IV Uncertain Knowledge and Reasoning**
- 13 Quantifying Uncertainty
- 14 Probabilistic Reasoning
- 15 Probabilistic Reasoning over Time
- 16 Making Simple Decisions
- 17 Making Complex Decisions

**Part V Learning**
- 18 Learning from Examples
- 19 Knowledge in Learning
- 20 Learning Probabilistic Models
- 21 Reinforcement Learning

**Part VII Communicating, Perceiving, and Acting**
- 22 Natural Language Processing
- 23 Natural Language for Communication
- 24 Perception
- 25 Robotics

**Part VIII Conclusions**
- 26 Philosophical Foundations
- 27 AI: The Present and Future
Exercise = Essay, details still open

- The course has one compulsory exercise.
- To pass the course you need to pass it
- The exercise is graded on scale 0 – 10, in addition the voluntary presentation yields up to 4 points
- Grade ≥ 1 means passing
- Topic of your own choice from the area of AI.
- Group/single? Returned by date?
  - Oral presentation 20 min

Grading

- In grading there are several components:
  - The exam (max 30 p.) May 18
  - An essay (max 10 p.) DL?
  - The max points altogether is 40 p.
  - Oral presentation (max 4 p.) After Vappu
  - Weekly exercises earn extra points (max 6 p.)

- Most probably the grade is decided as follows:

<table>
<thead>
<tr>
<th>points</th>
<th>20</th>
<th>24</th>
<th>28</th>
<th>32</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>grade</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Material

- The textbook of the course is

- There is no prepared material, the slides appear in the web as the lectures proceed

- The exam is based on the lectures (i.e., not on the slides only)

Weekly Exercises

- It is most advisable to take part in the weekly exercises
- The exercise questions appear in the web on Thu of previous week the latest
- Being ready to present one’s own solution to a question publicly yields one mark.
- Each session has c. 5 questions ⇒ you may gather altogether c. 5×12 = 60 marks

<table>
<thead>
<tr>
<th>Marks</th>
<th>Extra points</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>1</td>
</tr>
<tr>
<td>40%</td>
<td>2</td>
</tr>
<tr>
<td>50%</td>
<td>3</td>
</tr>
<tr>
<td>60%</td>
<td>4</td>
</tr>
<tr>
<td>70%</td>
<td>5</td>
</tr>
<tr>
<td>80%</td>
<td>6</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

- Artificial intelligence is a wide and far-reaching concept. It also keeps changing over time.
- Nowadays maybe more of a playground of philosophers and cognitive scientists.
- From the point of view of computer science AI comprises of a set of more focused research fields that have already drifted quite far apart from each other.
- The common goal in different subfields is to raise the "intelligence" of computers/machines.
  - I.e., to make the use of software easier.
- As a result one gets ready-to-use software and theory charting out the boundaries of mechanical computation.

1.1 What is AI?

- The only comparison to an intelligent machine that we are aware of is ourselves.
- On the other hand, comparison to the human intelligence limits out other (better) alternatives.
- Ideal intelligence is called rationality.
- One can view intelligence from the point of view of though and behavior.
- As combinations we get four distinct views to artificial intelligence: thinking humanly, acting humanly, thinking rationally, and acting rationally.
Turing Test

- English mathematician Alan Turing proposed in 1950 the following criterion for the intelligence of a machine: a human interrogator cannot differentiate whether s/he is communicating with another human or a computer using text messages.
- An example of a test of acting human-like.
- In the so-called total Turing test the machine also has to be able to observe and manipulate its physical environment.
- Time-limited Turing test competitions are organized annually.
- The best performance against knowledgeable organizers is recorded by programs that try to fool the interrogator.
- Human experts have the highest probability of being judged as non-humans.

Rationality

- Study of rational thought is essentially study of formal logic and logical deduction.
- Methods that are based (only) on logics suffer from computational complexity issues and the difficulty of expressing uncertain knowledge.
- In the model of rational acting we examine agents.
- An agent is something that acts.
- A software agent is distinguished from a program e.g. by:
  - its autonomous control,
  - capability to perceive its environment,
  - ability to adapt to change,
  - persistence over a prolonged time,
  - ability to take on another’s goals,
  - and so forth.
A rational agent works to reach the best possible outcome given its observations and knowledge.
Under uncertainty one aims at maximizing expectation over the outcome.
Rational performance on the long run may require one to perform seemingly irrationally for shorter periods.
In the real world total rationality is not usually possible (due to the lack of time).

1.4 The State of the Art

Different activities in many subfields:
- **Robotic vehicles**: Driverless robotic cars are being developed in closed environments and more and more in daily traffic. Modern cars recognize speed limits, adapt to the traffic, take care of pedestrian safety, can park themselves, have intelligent light systems, wake up the driver, …
- **Speech recognition**: Many devices and services nowadays understand spoken words (even dialogs)
- **Autonomous planning and scheduling**: E.g. space missions are tomorrow planned autonomously
- **Game playing**: Computers defeat human world champions in chess systematically and convincingly. Watson that won human players in Jeopardy was today named by HS as the science achievement of 2011
• **Spam fighting**: Learning algorithms reliably filter away 80% or 90% of all messages saving us time for more important tasks

• **Logistics planning**: E.g. military operations are helped by automated logistics planning and scheduling for transportation

• **Robotics**: Autonomous vacuum cleaners, lawn movers, toys, and special (hazardous) environment robots are common these days

• **Machine translation**: Translation programs based on statistics and machine learning are in ever increasing demand (in particular in EU)

There are of course many other interesting AI applications some of them taking advantage of the Web

---

2 INTELLIGENT AGENTS

• An agent perceives its **environment** through **sensors** and acts upon the environment through **actuators**
  • Our sensors include eyes, ears, nose, and other organs
  • Our actuators include hands, legs, mouth, and other body parts
  • The sensors of a robot (or a car) can include e.g. cameras, infrared and laser range finders and various motors as actuators
  • A software agent receives keystrokes, file contents, and network packets as sensory inputs
  • It acts on the environment by displaying on the screen, writing files, and sending network packets
• We assume that every agent will perceive its own actions, but not always the effects.
• In general, an agent’s choice of action at any given instant can depend on the entire percept sequence observed to date.
• The agent function maps any given percept sequence to an action.
• The table of all possible input-output pairs of the function is a complete external characterization of the agent.
• Of course such a table is infinite in most cases < not applicable.
• Internally the agent function for an artificial agent will be implemented by an agent program.

Measuring the Performance of an Agent

• The rational agent that we are aiming at should be successful in the task it is performing.
• To assess the success we need to have a performance measure.
• What is rational at any given time depends on:
  - The performance measure that defines the criterion of success.
  - The agent’s prior knowledge of the environment.
  - The actions that the agent can perform.
  - The agent’s percept sequence to date.

For each possible percept sequence a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.
• Junk mail filtering has to classify e-mail messages as junk or relevant messages
• A physician has to decide whether to operate on a patient or not
• The number of correctly classified instances is not the best possible measure of performance, because right and wrong decisions have different weights
• Moving an important message to the junk folder is worse than letting through some junk mails occasionally

<table>
<thead>
<tr>
<th>True Positive</th>
<th>False positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>False negative</td>
<td>True negative</td>
</tr>
</tbody>
</table>

Properties of Environments

Task environments can be classified at least by the following properties

- **Fully vs. partially observable**
  - In a fully observable environment, the agent’s sensors give it all relevant aspects affecting its choice of performance
  - Hence, the agent does not need to maintain any internal state (understanding of the state of the world)
  - An environment might be partially observable because of noisy and inaccurate sensors or simply due to its basic nature

- **Deterministic vs. stochastic**
  - If the next state of the environment is completely determined by the current state and the action executed, then we say that it is deterministic. Otherwise, it is stochastic.
  - A deterministic environment may appear stochastic if it is partially observable
Task Environments /2

- **Episodic vs. sequential**
  - In an episodic task environment, the next episode does not depend on the actions taken in previous episodes.
  - In sequential environments the current decision could affect all future decisions.

- **Static vs. dynamic**
  - In a static environment the agent may stop to deliberate its actions without fearing that the world changes.
  - In a dynamic environment the agent has to keep looking for the state of the environment.
  - In a *semidynamic* environment the environment itself does not change with the passage of time, but the agent’s performance score does (the agent is penalized for the time required to plan its actions).

Task Environments /3

- **Discrete vs. continuous**
  - The distinction can be made with respect to the state of the environment, time, and the percepts and actions of the agent.

- **Single agent vs. multiagent**
  - Depends on which entities one wants to view as agents or just simply as stochastically behaving objects.
  - In the multiagent environment one can compete or cooperate.

- **Known vs. unknown**
  - Whether the “laws of physics” of the environment are known.
  - In a known environment the outcomes (or outcome probabilities in a stochastic environment) for all actions are given.
  - A known environment can be partially observable.
Real World from the Perspective of a Robot

- Partially observable
  - Sensors are not perfect and perceive only the close environment (holds also for humans)
- Stochastic
  - Wheels slip, batteries run out, parts break — one can never be certain that the intended action is fulfilled
- Sequential
  - The effects of actions change over time → a robot has to manage sequential decision problems and be able to learn
- Dynamic
  - When to deliberate, when to act
- Continuous/Infinite
  - Algorithms must work in this environment, not, e.g., in a finite discrete search space
- Single/multiagent environment
  - Depending on whether one wants to look at other objects as agents or stochastic parts of the environment

2.4 The Structure of Agents

- Because the agent’s history of percept-action pairs stored in a table describes the external behavior of the agent, in principle the control program of the agent could be based on table lookup
- Obviously, this only works for very small environments
- In more realistic situations tabulating percept-action pairs is not a viable solution
- Instead, the agent program has to be able to decide the desired action on any percept history without tabulating all possible alternatives
- The following agent types are the most common solutions to this problem
Simple / Model-based Reflex Agents

- The simplest possible control program makes the agent operate solely on the current percept discarding the percept history.
- Reflexes are used in emergency situations by humans as well as by robots.
- Reflexive behavior yields correct decisions only when the task environment is fully observable.
- One can choose a set of current rules based on the agent's internal state.
- Hence, one can maintain a model of the world covering some information that is not directly observable.

2.4.4 Goal-based Agents

- In addition to its percepts, the agent possesses knowledge of its goal.
- The goal is some assertion concerning the environment which should be satisfied.
- By combining the goal and knowledge of the effects of available actions, the agent can try to satisfy the goal.
- If the goal cannot be satisfied directly through one action, one has to find out a sequence of actions to satisfy it.
- The agent may resort to search algorithms (next section) or planning (later on).
2.4.5 Utility-based Agents

- The agent may achieve its goal in many different ways – different solutions may have differences in quality
- Setting a goal alone does not suffice to express more complex settings
- If the possible states of the environment are assigned an order through an utility function, then the agent can try to improve its value
- In partially observable and stochastic environments we choose the action that maximizes the expected utility of the outcomes
- The utility function maps a state (or a sequence of states) onto a real number
- In order to take advantage of this approach, the agent does not necessarily have to possess a utility function explicitly

2.4.6 Learning Agents

- Programming agents for all possible tasks by hand appears to be a hopeless task
- Already Turing (1950) proposed machine learning as a method of creating intelligent systems
- Learning allows the agent to operate in initially unknown environments and become more competent than its initial knowledge alone might allow
- The learning element of an agent has to be kept distinct from the actual performance element

- We come back to the techniques of machine learning towards the end of the course
3 SOLVING PROBLEMS BY SEARCHING

- A goal-based agent aims at solving problems by performing actions that lead to desirable states
- Let us first consider the uninformed situation in which the agent is not given any information about the problem other than its definition
- In blind search only the structure of the problem to be solved is formulated
- The agent aims at reaching a goal state
- The world is static, discrete, and deterministic

- The possible states of the world define the state space \( \Sigma \) of the search problem

In the beginning the world is in the initial state \( s_1 \in \Sigma \)
- The agent aims at reaching one of the goal states \( G \subseteq \Sigma \)

Quite often one uses a successor function \( S: \Sigma \rightarrow P(\Sigma) \) to define the agent’s possible actions
- Each state \( s \in \Sigma \) has a set of legal successor states \( S(s) \) that can be reached in one step
- Paths have a non-negative cost, most often the sum of costs of steps in the path
The definitions above naturally determine a directed and weighted graph.

The simplest problem in searching is to find out whether any goal state can be reached from the initial state $s_1$.

The actual solution to the problem, however, is a path from the initial state to a goal state.

Usually one takes also the costs of paths into account and tries to find a cost-optimal path to a goal state.

Many tasks of a goal-based agent are easy to formulate directly in this representation.

For example, the states of the world of 8-puzzle (a sliding-block puzzle) are all $9! / 2 = 181440$ reachable configurations of the tiles.

Initial state is one of the possible states.

The goal state is the one given on the right above.

Possible values of the successor function are moving the blank to left, right, up, or down.

Each move costs one unit and the path cost is the total number of moves.
Donald Knuth’s (1964) illustration of how infinite state spaces can arise

**Conjecture**: Starting with the number 4, a sequence of factorial, square root, and floor operations will reach any desired positive integer

\[
\sqrt[3]{\sqrt{\sqrt{(4!)!}}} = 5
\]

- **States** \( \mathcal{S} \): Positive numbers
- **Initial state** \( s_0 \): 4
- **Actions**: Apply factorial, square root, or floor operation
- **Transition model**: As given by the mathematical definitions of the operations
- **Goal test**: State is the desired positive integer

**Search Tree**

- When the search for a goal state begins from the initial state and proceeds by steps determined by the successor function, we can view the progress of search in a tree structure
- When the root of the search tree, which corresponds to the initial state, is expanded, we apply the successor function to it and generate new search nodes to the tree — as children of the root — corresponding to the successor states
- The search continues by expanding other nodes in the tree respectively
- The search strategy (search algorithm) determines in which order the nodes of the tree are expanded
The node is a data structure with five components:

- The state to which the node corresponds,
- Link to the parent node,
- The action that was applied to the parent to generate this node,
- The cost of the path from the initial state to the node, and
- The depth of the node

Global parameters of a search tree include:

- $b$ (average or maximum) branching factor,
- $d$ the depth of the shallowest goal, and
- $m$ the maximum length of any path in the state space