CSC 480: Artificial Intelligence

- The materials in this course are modified and adapted from the following. The authors and their institutions have made these materials available with an Open Courseware License or for use for educational purposes only.

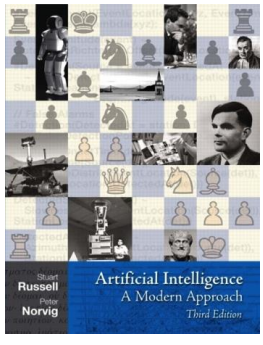
Sources:

- **Harvard's CS50: Introduction to Artificial Intelligence with Python**
  Brian Yu and David J. Malan
- **UCB's course materials from CS 188: Introduction to Artificial Intelligence**
  Dan Klein, Pieter Abbeel University of California, Berkeley
- **Cal Poly, San Luis Obispo CS 480:** Franz Kurfess

- **Artificial Intelligence: A Modern Approach**

Textbook

- Not required, but for students who want to read more we recommend: Russell & Norvig, AI: A Modern Approach



Motivation

- scientific curiosity
  - try to understand entities that exhibit intelligence
- engineering challenges
  - building systems that exhibit intelligence
- some tasks that seem to require intelligence can be solved by computers
- humans may be relieved from tedious tasks
- current hype

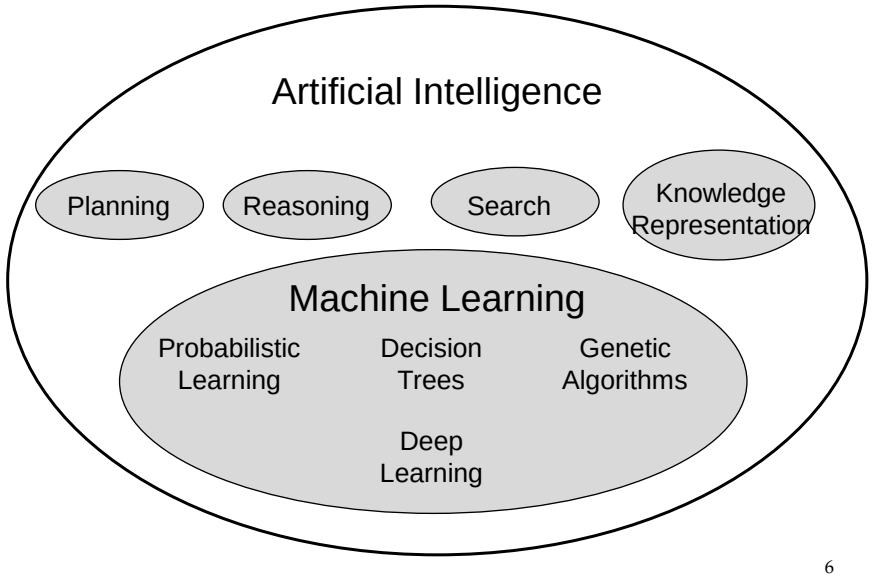
Today

- What is artificial intelligence?
- Where did it come from?
  - Intellectual Context
  - History
- What is this course?

What is AI?
Examples of Definitions

The science of making machines that:

- cognitive approaches (think like humans)
  - emphasis on the way systems work or "think"
  - requires insight into the internal representations of environment and rationality
- behavioral approaches (act like humans)
  - only observed behavior is taken into account
- rational systems
  - systems that do the "right thing"
  - idealized concept of intelligence

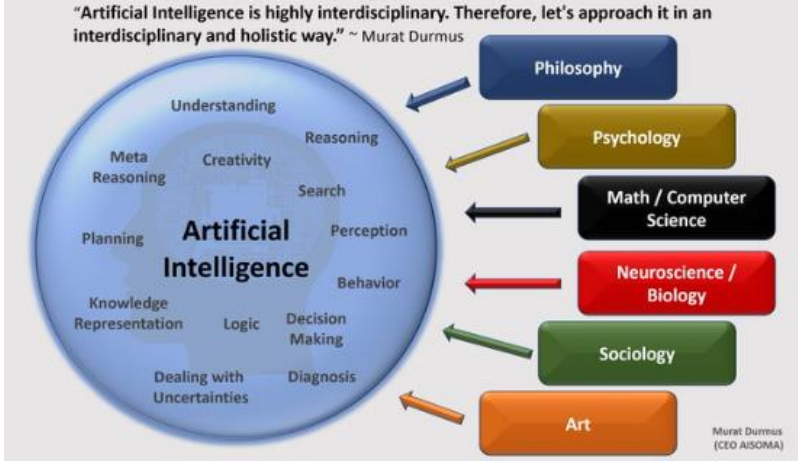


Rebooting AI : Gary Marcus

Outline of the Course

- Agents, State Spaces, and Search
- Game theory: Adversarial Search
- Logic and Inference, Knowledge Engineering
- Optimization and Constraint Satisfaction
- Uncertainty, Bayesian Networks
- Learning
- Neural Nets, Deep Learning

AI draws from multiple sources and contributes to multiple disciplines and applications

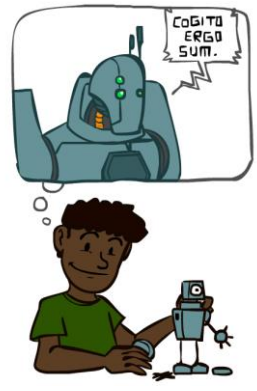


The Turing Test

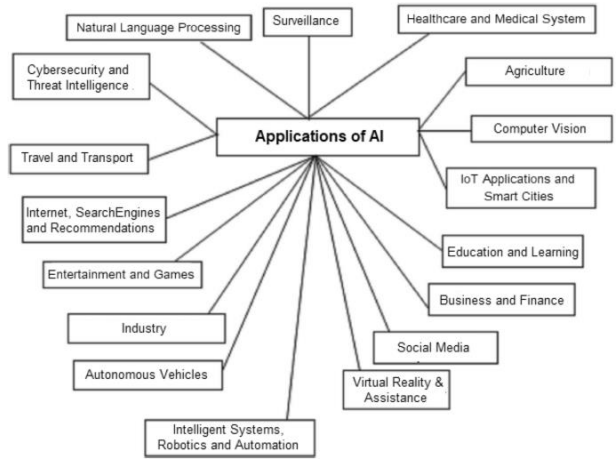
- proposed by Alan Turing in 1950 to provide an operational definition of intelligent behavior
  - the ability to achieve human-level performance in all cognitive tasks, sufficient to fool an interrogator
  - the *computer* is interrogated by a human via a teletype
  - *computer* passes the test if the interrogator cannot identify the answerer as computer or human
  - assessment usually through a jury instead of a single interrogator
- Concept has evolved and refined over time.
  - Many levels of measuring "intelligence" e.g. Strong vs Weak AI
  - Many controversies

A (Short) History of AI

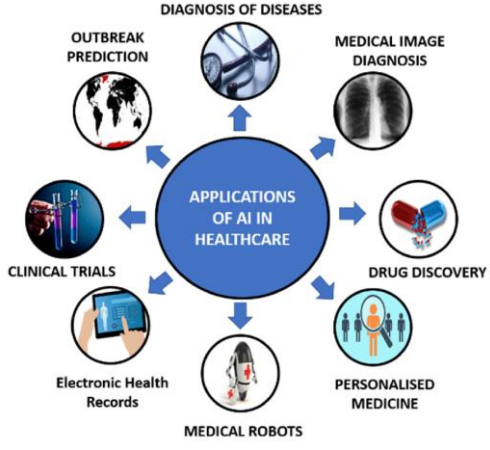
- 1940-1950: Early days
  - 1943: McCulloch & Pitts: Boolean circuit model of brain
  - 1950: Turing's "Computing Machinery and Intelligence"
- 1950—70: Excitement: Look, Ma, no hands!
  - 1950s: Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
  - 1956: Dartmouth meeting: "Artificial Intelligence" adopted
  - 1965: Robinson's complete algorithm for logical reasoning
  - First AI winter
- 1970—90: Knowledge-based approaches
  - 1969—79: Early development of knowledge-based systems
  - 1980—88: Expert systems industry booms
  - 1988—93: Expert systems industry busts: "AI Winter"
- 1990—: Statistical approaches
  - Resurgence of probability, focus on uncertainty
  - General increase in technical depth
  - Agents and learning systems… "AI Spring"?
- 2000—: Where are we now?
  - Machine learning, Neural Nets, Deep Learning (LLM)
  - Convergence of computational power and "Big Data sets"



Application Areas



Healthcare



AI Infrastructure

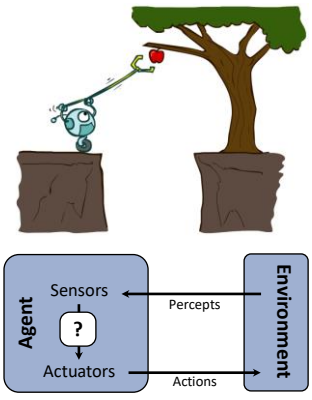
- Much better access to
  - libraries
  - frameworks
  - services
  - platforms
  - data
- Examples
  - Google's TensorFlow, AutoML
  - Facebook's PyTorch
  - Amazon's ML platform, Lex conversational framework, Polly text to speech service, Recognition image processing service
  - OpenAI Gym
    - » toolkit for developing and comparing reinforcement learning algorithms

Agents

- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

Designing Rational Agents

- An agent is an entity that perceives and acts.
- A rational agent selects actions that maximize its (expected) utility.
- Characteristics of the percepts, environment, and action space dictate techniques for selecting rational actions
- This course is about:
  - General AI techniques for a variety of problem types
  - Learning to recognize when and how a new problem can be solved with an existing technique



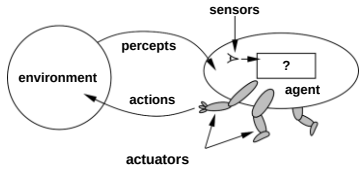
Rational Decisions

- We'll use the term rational in a very specific, technical way:
  - Rational: maximally achieving pre-defined goals
  - Rationality only concerns what decisions are made (not the thought process behind them)
  - Goals are expressed in terms of the utility of outcomes
  - Being rational means maximizing your expected utility **given current knowledge**

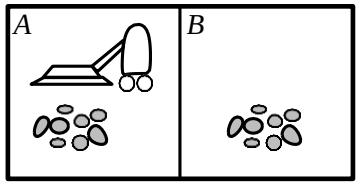
Course Focus in on ***Computational Rationality***

Agents and environments

- Agents include humans, robots, softbots, thermostats, etc.
- An agent can be anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
- The agent function maps from percept histories to actions: Mapping : $P^* \rightarrow A$
- The agent program runs on the physical architecture to produce Mapping



Vacuum-cleaner world



- Percepts: location and contents, e.g., [*A, Dirty*]
- Actions: *Left, Right, Suck, NoOp*

A vacuum-cleaner agent

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
.	.

```
function Reflex-Vacuum-Agent( [location,status]) returns an action
    if status = Dirty then return Suck else
    if location = A then return Right else
    if  location = B then return Left
```

What is the right function?
Can it be implemented in a small agent program?

Rationality (acts to maximize utility)

- Fixed performance measure evaluates the environment sequence
  - one point per square cleaned up in time T ?
  - one point per clean square per time step, minus one per move?
  - penalize for > k dirty squares?
- A **rational agent** chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date

Rationality

- Note that rationality is **not** the same as omniscience; an omniscient agent knows the actual outcome of its actions and can act accordingly. Percepts may not supply all relevant information.
- Similarly, rationality is **not** the same thing as clairvoyance (action outcomes may be unexpected) nor perfection (we maximize expected performance, not actual performance).
- Performing actions to modify future percepts (i.e. information gathering) is a crucial part of rationality and is closely aligned with exploration.

Task Environment: Designing a rational agent

- Specify the task environment
- Example, designing an automated taxi:
  - Goals ⇒ P: Performance measure??
  - Model ⇒ E: Environment??
  - Actions ⇒ A: Actuators??
  - Information ⇒ S: Sensors??

PEAS

- To design a rational agent, we must specify the task environment
- Consider, e.g., the task of designing an automated taxi:
  - Performance measure??
    - » safety, destination, profits, legality, comfort, . . .
  - Environment??
    - » US streets/freeways, traffic, pedestrians, weather, . . .
  - Actuators??
    - » steering, accelerator, brake, horn, speaker/display, . . .
  - Sensors??
    - » video, accelerometers, gauges, engine sensors, keyboard, GPS, . . .

Environment types

- Observable: fully vs partial
  - fully: sensors give access to state of complete environment at any point in time
- Deterministic: deterministic vs non-deterministic (≈ stochastic)
  - Next state of the environment is completely determined by current state and action
- Static: static vs dynamic
  - Dynamic is environment changes while agent is deciding what action to take
- Agents: Single vs multi-agents

- Discrete: discrete vs continuous
  - Model of state, time, precepts, actions
- Episodic: episodic vs sequential
  - Agent's experience is divided into atomic groups of steps

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?? Deterministic?? Static?? Single Agent??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Static??	Yes	Semi	Semi	No
Single Agent??	Yes	No	No	No

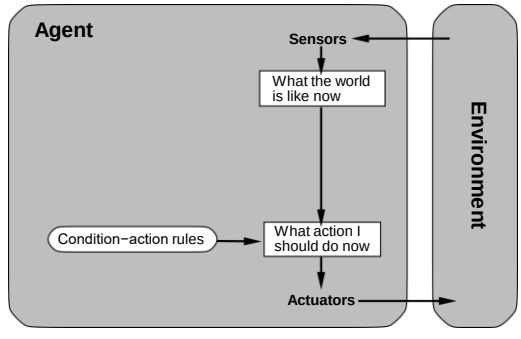
- The environment type largely determines the agent design
- The real world is (of course)
  partially observable, stochastic, dynamic, multiple agents

Some Types of Agents

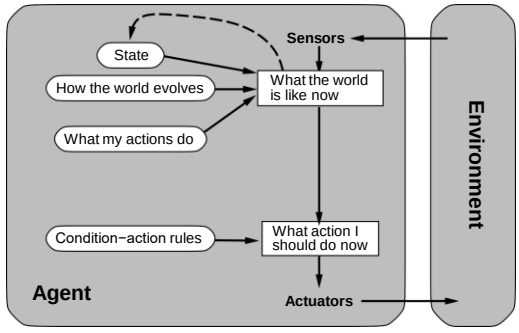
https://www.simform.com/blog/types-of-ai-agents/

- Simple Reflex Agent:
  - An agent system that follows pre-defined rules to make decisions using only predefined rules without considering the past.
- Model or State based Agent
  - Maintains internal state that has information about the world
- Goal-based Agent:
  - Use information from their environment to achieve specific goals. They employ search algorithms to find the most efficient path towards their objectives within a given environment.
- Utility-based Agents
  - Make decisions based on maximizing a utility function or value. They choose the action with the highest expected utility, which measures how good the outcome is.
- Learning Agents
  - agent that can learn from past experiences and improve its performance.
  - Can apply to all of the above

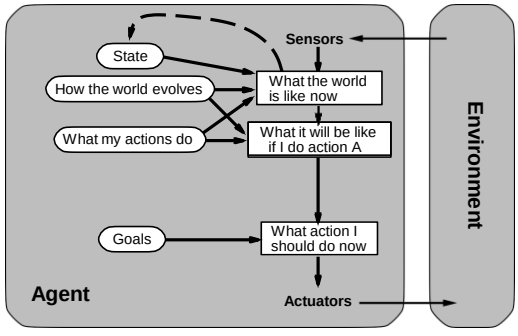
Simple reflex agents



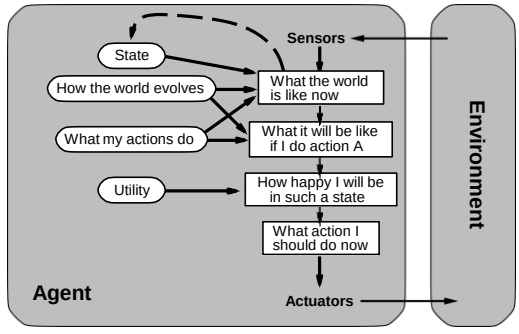
Reflex agents with state



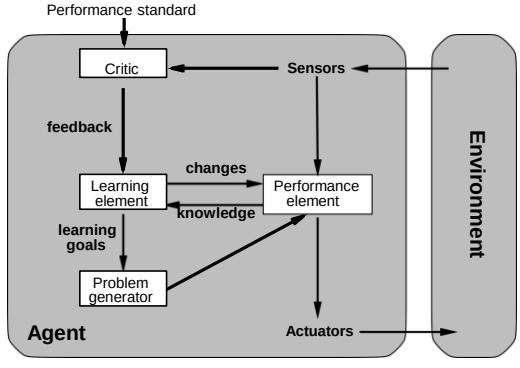
Goal-based agents



Utility-based agents



Learning agents



Summary: Key Takeaways

- Computational Rationality
- Classic Turing Test
- The agent function describes what the agent does in all circumstances The performance measure evaluates the environment sequence
- Types of Agents:
  - reflex, reflex with state, goal-based, utility-based
- PEAS descriptions define task environments
- Environments are categorized along different dimensions:
  - observable? deterministic? episodic? static? discrete? single-agent?

Extra Slides An Intelligent Agent

- An intelligent agent should not only gather information, but also learn.
- The agent's initial configuration could reflect some prior knowledge of the environment, but as the agent gains experience, this may be modified and augmented
- Generally speaking, a rational agent should be autonomous, in the sense that it learns what it can to compensate for partial or incorrect prior knowledge. After sufficient experience of its environment, the behavior of a rational agent can become effectively independent of its prior knowledge.
- Ideally, the incorporation of learning allows for the design of a single rational agent that will succeed in a variety of different environments and for a variety of tasks.

Asymptotic Complexity: Why

Problem: The likely performance or space requirements of an algorithm or comparing two algorithms

Asymptotic Notation: What is needed

- A theoretical framework that can be applied to a wide range of algorithms
  - O(), Θ(), Ω() notations (Asymptotic growth rate)
  - Ignore lower order terms (irrelevant for large problem sizes) and constant multipliers (system dependent)
- A procedure to calculate a "metric of performance"
- A method to determine what problems are:
  - Unable to have a practical algorithmic solution and measure the hardness of such problems
    - » NP-complete, NP-hard, ….

Asymptotic Order of Growth

- Idea: Order of growth within a constant multiple as n→∞
- Simple questions this can help us answer:
  - How much faster will algorithm run on computer that is twice as fast?

  vs.
  - How much longer does it take to solve problem of double input size?
- The actual times vary due to many factors but asymptotic analysis provides a quick and dirty way to compare algorithms and guidance in where to look to improve performance!

Why order of growth matters!

	n	$n \log_2 n$	n^2	n^3	1.5^n	2^n	$n!$
$n = 10$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	4 sec
$n = 30$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	18 min	10^{25} years
$n = 50$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	11 min	36 years	very long
$n = 100$	< 1 sec	< 1 sec	< 1 sec	1 sec	12,892 years	10^{17} years	very long
$n = 1{,}000$	< 1 sec	< 1 sec	1 sec	18 min	very long	very long	very long
$n = 10{,}000$	< 1 sec	< 1 sec	2 min	12 days	very long	very long	very long
$n = 100{,}000$	< 1 sec	2 sec	3 hours	32 years	very long	very long	very long
$n = 1{,}000{,}000$	1 sec	20 sec	12 days	31,710 years	very long	very long	very long

These are old numbers but note that to get a very rough approximation for today's higher performance computers divide the time by a factor of 10 or more. But
note that:

- This makes little difference for n^3 or exponential or factorial performance for problem sizes over 1,000
- Historically improvements in software performance have had a larger impact for many applications rather than hardware improvements.

Formal Definitions: Asymptotic Order of Growth

- **T(n) is O(f(n))** if there exist constants C > 0 and $n_0 \geq 0$ such that $T(n) \leq C \cdot f(n)$ for all $n \geq n_0$.
  O(f(n)) is used to specify an upper bound.
- **T(n) is Ω(f(n))** if there exist constants C > 0 and $n_0 \geq 0$ such that $T(n) \geq C \cdot f(n)$ for all $n \geq n_0$.
  Ω(f(n)) is used to specify a lower bound.
- **T(n) is Θ(f(n))** if T(n) is both O(f(n)) and Ω(f(n)) or equivalently if there exist constants $C_1 > 0$, $C_2 > 0$ and $n_0 \geq 0$ such that $C_1 \cdot f(n) \leq T(n) \leq C_2 \cdot f(n)$ for all $n \geq n_0$
  Θ(f(n)) is used to specify a tight bound.

Asymptotic Complexity Classes: O(g(n)), Ω(g(n)), θ(g(n)

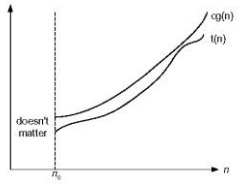


Figure 2.1 Big-oh notation: $t(n) \in O(g(n))$

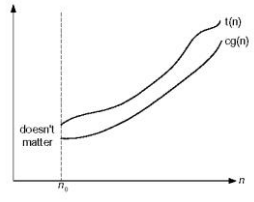


Fig. 2.2 Big-omega notation: $t(n) \in \Omega(g(n))$

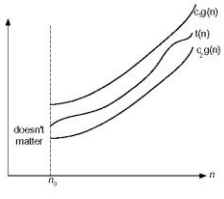


Figure 2.3 Big-theta notation: $t(n) \in \Theta(g(n))$

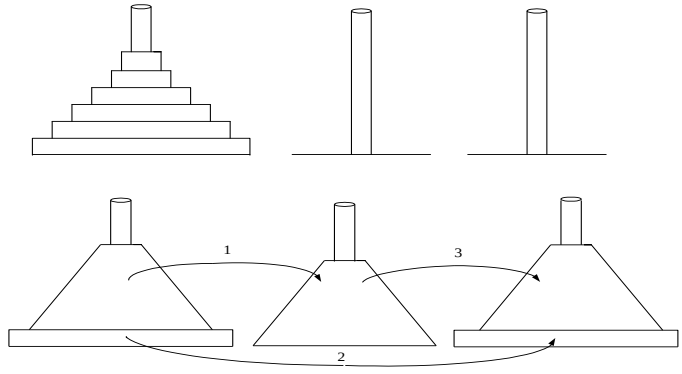
Orders of growth of some important functions

- All logarithmic functions $\log_a n$ belong to the same class $\Theta(\log n)$ no matter what the logarithm's base $a > 1$ is
- All polynomials of the same degree k belong to the same class: $a_k n^k + a_{k-1} n^{k-1} + \ldots + a_0 \in \Theta(n^k)$
- Exponential functions a^n have different orders of growth for different a's, e.g. 3^n grows faster then 2^n
- order log n < order n^α (α>0) < order a^n < order n! < order n^n

Important Orders of Growth: $\theta(n)$ // in order!

Logarithmic:	$\theta(\log n)$	binary search
Linear :	$\theta(n)$	merging 2 lists
Log-linear :	$\theta(n \log n)$	merge sort
Quadratic :	$\theta(n^2)$	counting pairs
Cubic :	$\theta(n^3)$	counting triples, matrix multiply
Exponential :	$\theta(2^n)$	searching all subsets
Factorial :	$\theta(n!)$	searching all permutations

The Tower of Hanoi Puzzle



Idea translates naturally to pseudo code for recursive function

```
1. Han (n, src, targ, aux)        // n = number of disks
2.       if n>=1
3.                Han (n-1, src, aux, targ)
4.                Move nth disk from src to targ
5.                Han (n-1, aux, targ, src)
```

And this code translates naturally to a recurrence relation for counting "Moves"

M(n) = number of moves required for n-disks

M(n) = # a moves done in line 3 + 1 + # moves done in line 5

= M(n-1)+1+M(n-1) = 2*M(n-1) + 1

Optional Slides

Philosophy

- related questions have been asked by Greek philosophers like Plato, Socrates, Aristotle
- theories of language, reasoning, learning, the mind
- dualism (Descartes)
  - a part of the mind is outside of the material world
- materialism (Leibniz)
  - all the world operates according to the laws of physics

Mathematics and Statistics

- formalization of tasks and problems
- logic
  - propositional logic
  - predicate logic
- computation
  - Church-Turing thesis
  - intractability: NP-complete problems
- probability
  - degree of certainty/belief

Psychology

- behaviorism
  - only observable and measurable percepts and responses are considered
  - mental constructs are considered as unscientific
    - » knowledge, beliefs, goals, reasoning steps
- Neuroscience and cognitive psychology
  - the brain stores and processes information
  - cognitive processes describe internal activities of the brain
  - computational models of these processes lead to conjectures about the human brain and how it works

Computer Science

- Computational Models
  - Church-Turing thesis
  - intractability: NP-complete problems
- provides tools for testing theories
- programmability
- Speed, storage, actions (internet as data source)

Linguistics

- understanding and analysis of language
  - sentence structure, subject matter, context
- knowledge representation
- computational linguistics, natural language processing
  - hybrid field combining AI and linguistics