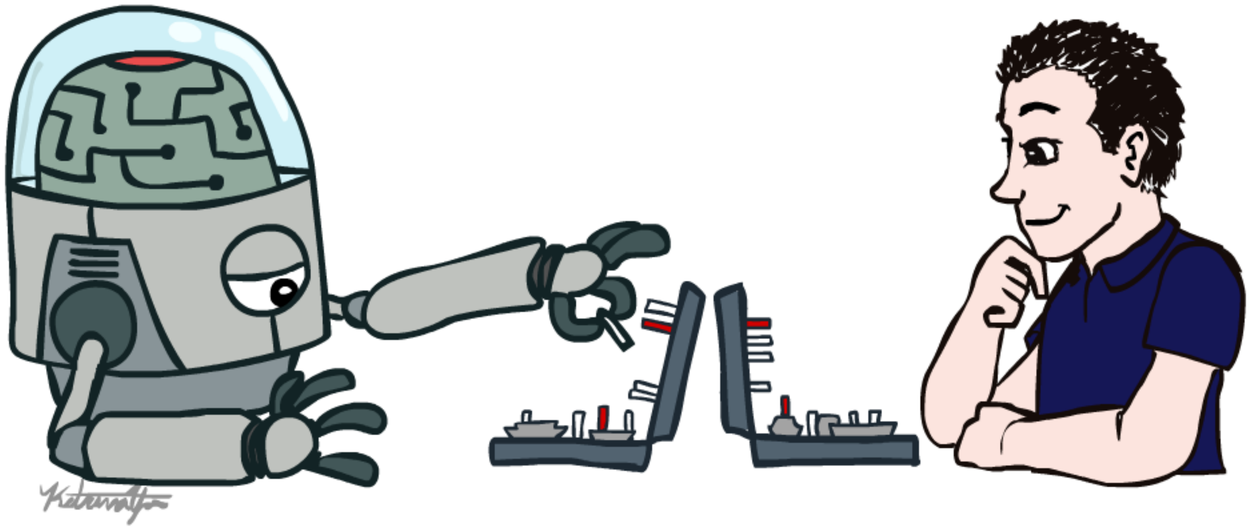
# CS383: Artificial Intelligence

## Introduction



Prof. Scott Niekum

University of Massachusetts Amherst

[Based in part on slides created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

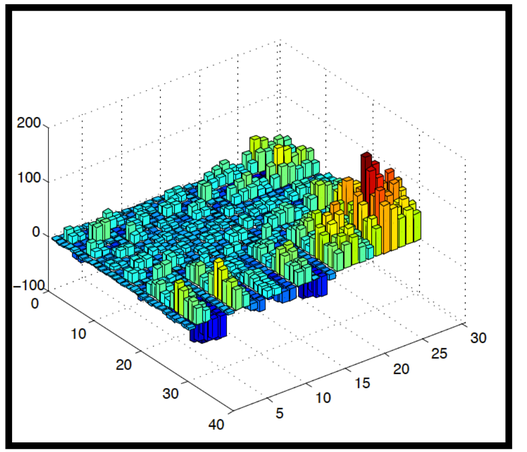
All original materials available at http://ai.berkeley.edu.]

# A bit about me

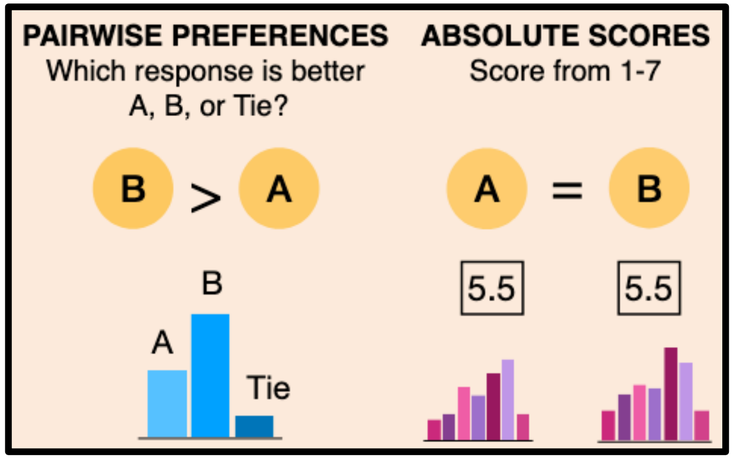
AI Safety

SCALAR LAB

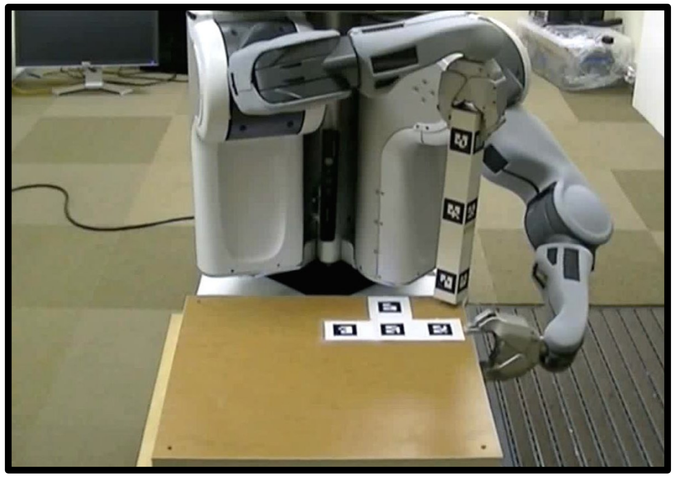
Safe, Confident, and Aligned Learning + Robotics




Reinforcement learning




LLMs and alignment



Robotics and imitation learning

# Course Information

- Communication:
  - Announcements on Canvas/Piazza
  - Grades on Gradescope
  - Piazza for discussion

- Assignments:
  - Gradescope for interactive homework (unlimited submissions!)
  - Autograded programming projects (submit via Gradescope)
  - Make sure you have access to a system where you can run Python

Class website:

https://people.cs.umass.edu/~sniekum/classes/383-F25/desc.php

(Link available on class Canvas page)

**TAs:**
Sankaran Vaidyanathan (sankaranv@cs.umass.edu)
Fareya Ikram (fikram@umass.edu)

**TA Office Hours:**
Tuesdays 1-3pm in CS207
Fridays 3-5pm in CS207

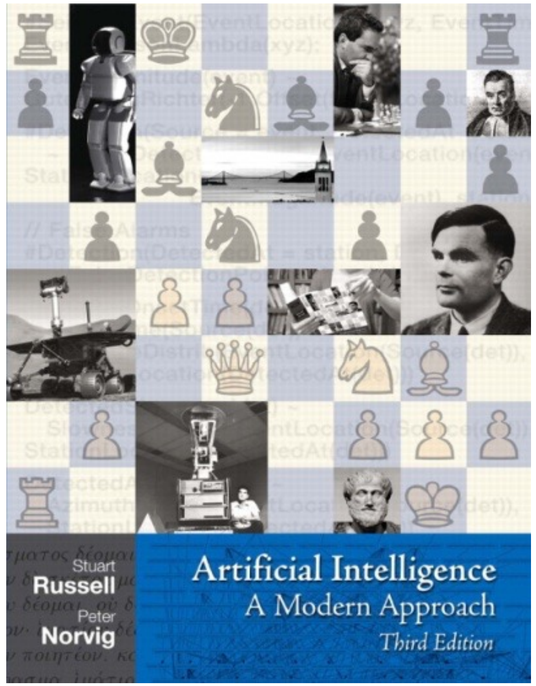
And by appointment, when needed

# Workload

- There will be a lot of math (and programming)
- Reading assignments
- 8 homework assignments:
  - ~2 weeks for each, but sometimes overlapping
  - Online, autograded, solve and submit alone
  - Can be turned in up to 5 days late for -20%
- 5 programming projects
  - Python, groups of 1 or 2 (except Project 0)
  - ~2 weeks for each, non-overlapping
  - Can be turned late until last day of classes for -20%
- One midterm, one final

# Textbook

Russell & Norvig, AI: A Modern Approach, 3rd Ed.

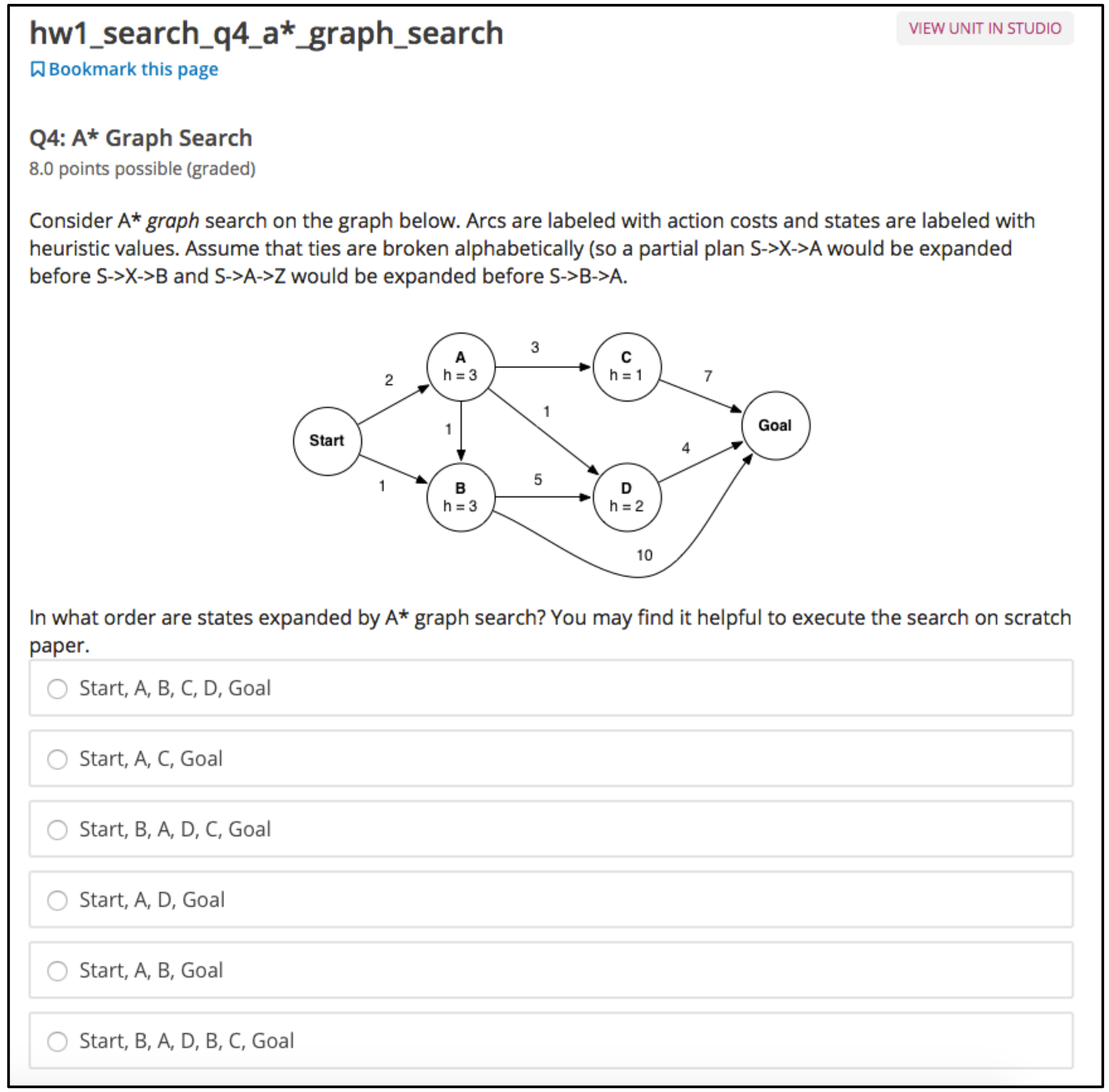



- I'll also post class slides

Warning: Not a course textbook, so our presentation does not necessarily follow the presentation in the book.

# Homework Exercises

- Online on Gradescope
- Autograded text boxes / multiple choice
- Try as many times as you want!
- Goal: self-assess and prepare for tests
- Can discuss at high-level, but work alone
- No spoilers on Piazza discussions!



**hw1_search_q4_a*_graph_search** VIEW UNIT IN STUDIO

Bookmark this page

**Q4: A* Graph Search**

8.0 points possible (graded)

Consider A* *graph* search on the graph below. Arcs are labeled with action costs and states are labeled with heuristic values. Assume that ties are broken alphabetically (so a partial plan S->X->A would be expanded before S->X->B and S->A->Z would be expanded before S->B->A.

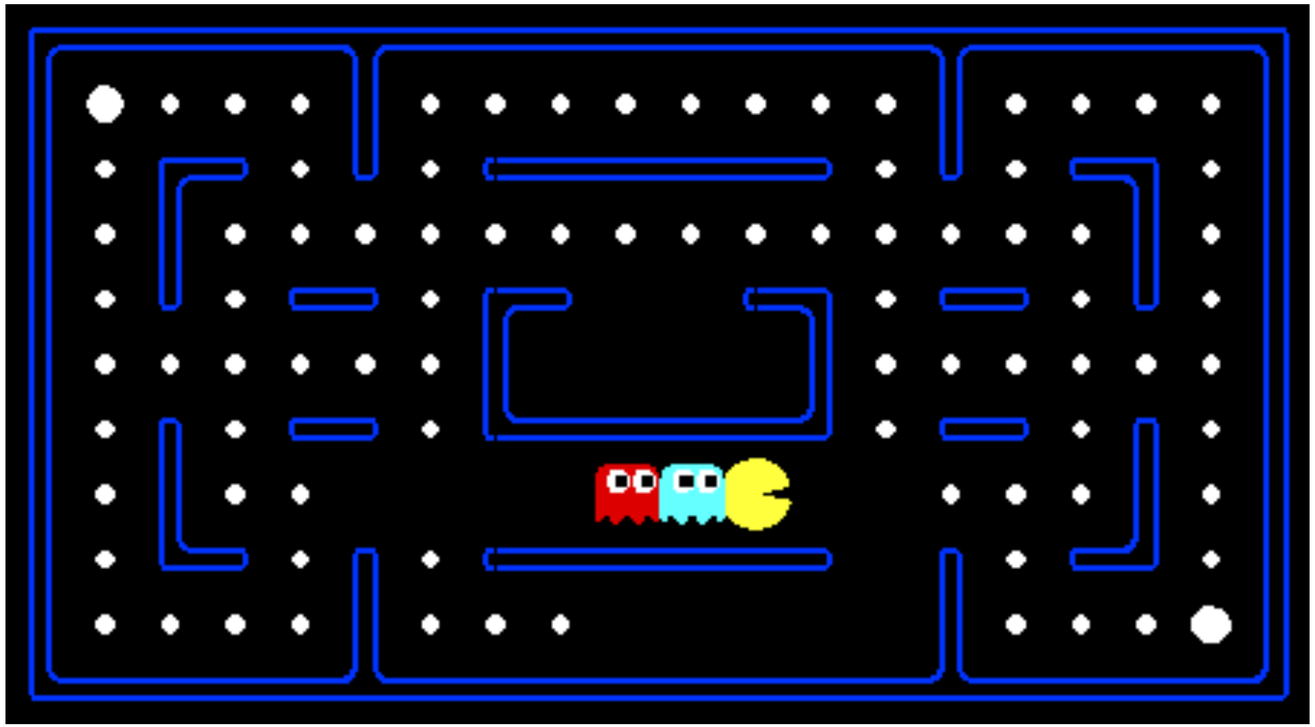


In what order are states expanded by A* graph search? You may find it helpful to execute the search on scratch paper.

- Start, A, B, C, D, Goal
- Start, A, C, Goal
- Start, B, A, D, C, Goal
- Start, A, D, Goal
- Start, A, B, Goal
- Start, B, A, D, B, C, Goal

# Programming Assignments

Pacman domain



Projects include:

- path planning and search
- multi-agent game trees
- reinforcement learning
- state estimation

Highly suggested: Pair programming
(switch “driver” and “observer” roles often)

# Midterm and Final

- Midterm will cover roughly half the class material
- Final will be comprehensive
- Midterm in-class, Final during finals week
- Very similar to Gradescope homework questions

# Attendance

- We will use Qwickly to take attendance — bring a device such as a smartphone that can read QR codes.
- You get 3 free skips, but save them for necessities, as no other absences will be excused, aside from truly exceptional circumstances.

# Grading

Plus/minus grading:

93-100: A
90-93: A-
87-90: B+
83-87: B
80-83: B-
77-80: C+
73-77: C
70-73: C-
67-70: D+
63-67: D
60-63: D-
<60: F

Grades will be weighted as follows:

- Attendance (10%)
- Gradescope Exercises (25%)
- Programming Assignments (35%)
- Midterm (15%)
- Final (15%)

# Academic Honesty

**READ THE STATEMENT IN THE SYLLABUS**

- Discuss concepts, but don’t share solutions or written work with other students
- Don’t look for answers / code online or elsewhere
- **No use of AI tools, such as ChatGPT** for homework or programming assignments
- Automated tools will be used to discover cheating
- If unsure, check university guidelines or ask — ignorance is not an excuse
- We will pursue the harshest penalties available, so please don’t cheat!
- To be clear: you will fail the class automatically and be reported to the university

# Important This Week

- Important this week:
  - **Make sure** that you can get into Gradescope — create an account and use the course code that you were emailed (and which is also posted on Piazza).
  - **Be sure** that you have a computer where you can run Python
  - **P0: Python tutorial** is out (due 9/8 at 11:59 pm via Gradescope)
- Also important:
  - **If you are wait-listed,** you might or might not get in depending on how many students drop. Be patient if possible — many students often drop early in the course.
  - **Office Hours** begin Friday

# A definition for AI

# A definition for AI


“Artificial Intelligence (AI) is a science and a set of computational technologies that are inspired by — but typically operate quite differently from — the ways people use their nervous systems and bodies to sense, learn, reason, and take action.”

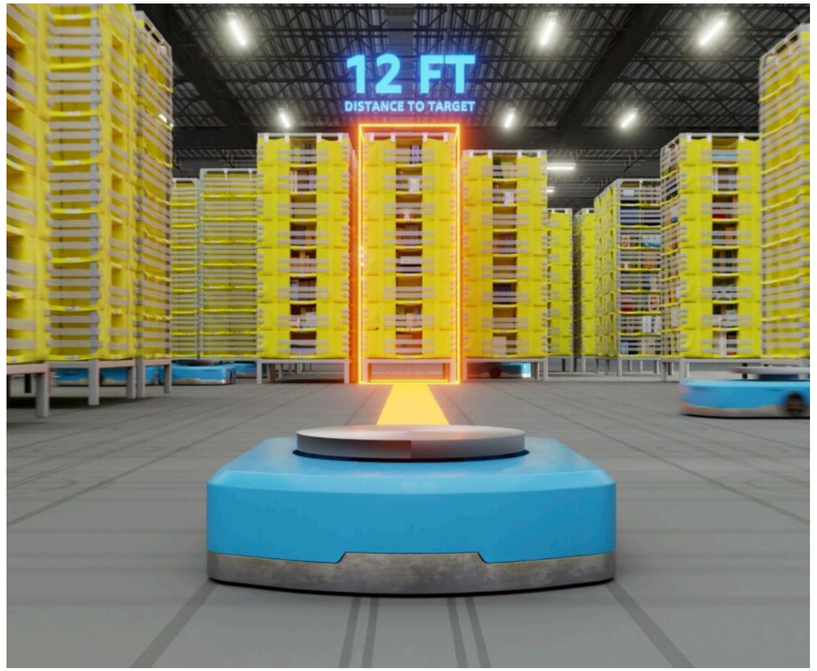
# This is not a deep learning course!

…but it is highly relevant to understanding current deep learning research and broader AI systems

- AI is bigger than just deep learning
- Foundational AI techniques are still used in modern research and practice
- These topics are also critical for understanding deep learning
- …and are often combined with deep learning

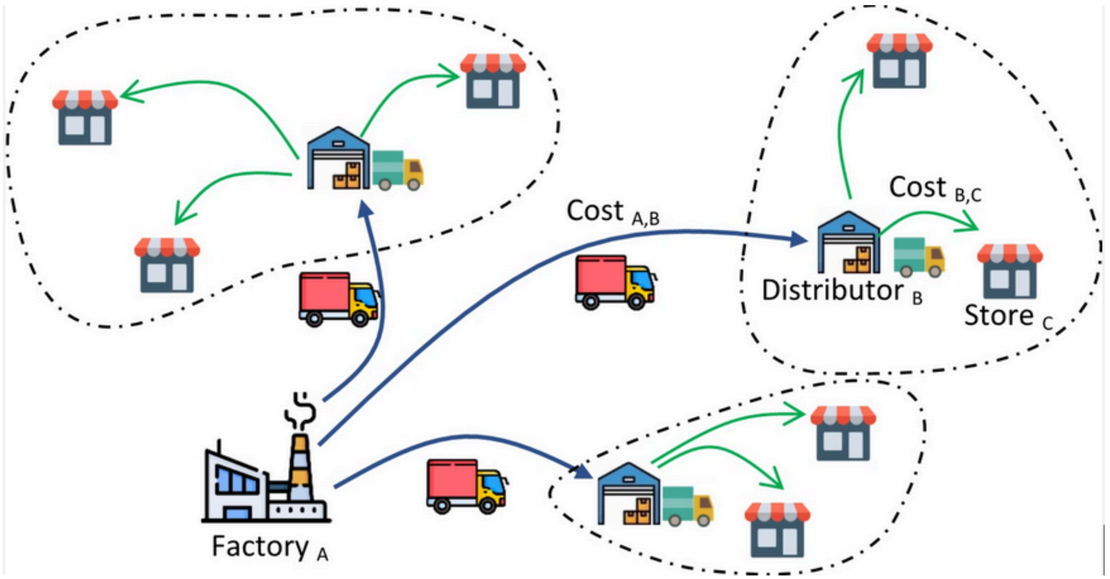
# Example: Search methods

- Modern path planning still widely employs traditional search methods, such as A*
- Top-k sampling and beam search are used for LLM text generation and are variants of greedy search and UCS
- Game trees + search are the foundation of MCTS, which powers systems such as AlphaGo (alongside a deep learning component)



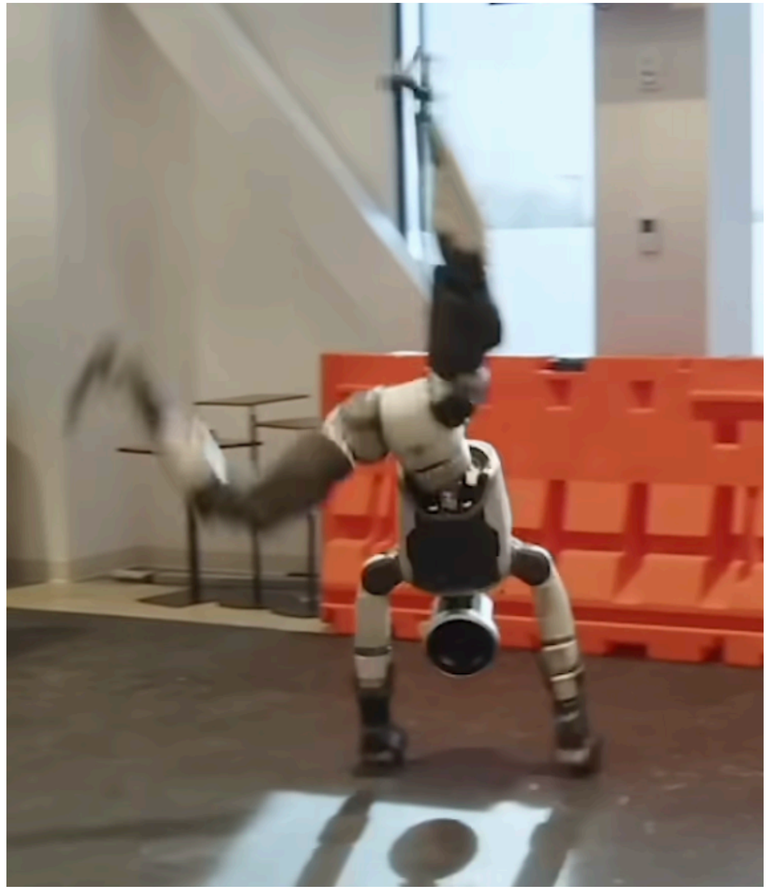
# Example: Constraint satisfaction

- Large-scale logistics problems often modeled as CSPs
- Interesting connections to Bayes nets
- Combinatorially hard problems can be solved approximately by gradient descent!
- Recent neural trajectory planners use CSPs to find collision-free, safe plans
  - e.g. https://arxiv.org/html/2503.19466v1



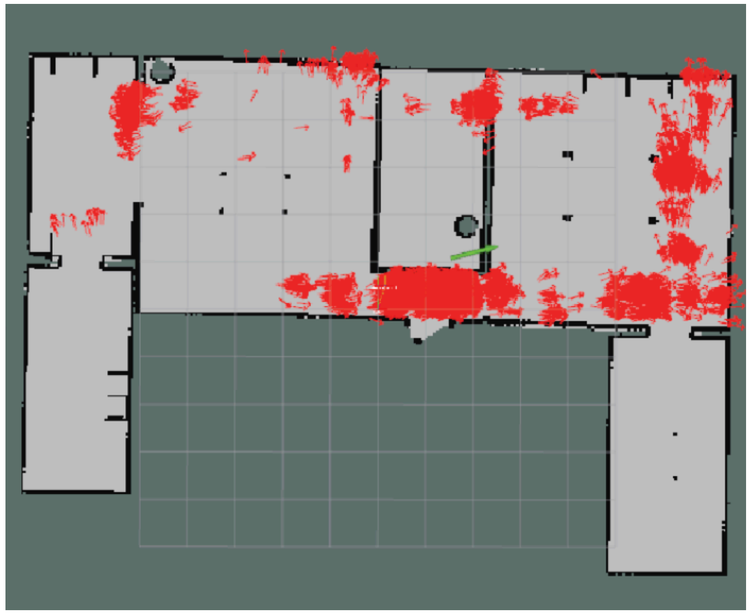
# Example: Reinforcement learning

- LLMs are “aligned” via RLHF
- AlphaGo also used techniques from RL
- Robot manipulation, parkour, soccer
- Most modern RL methods use neural network policies and/or value functions



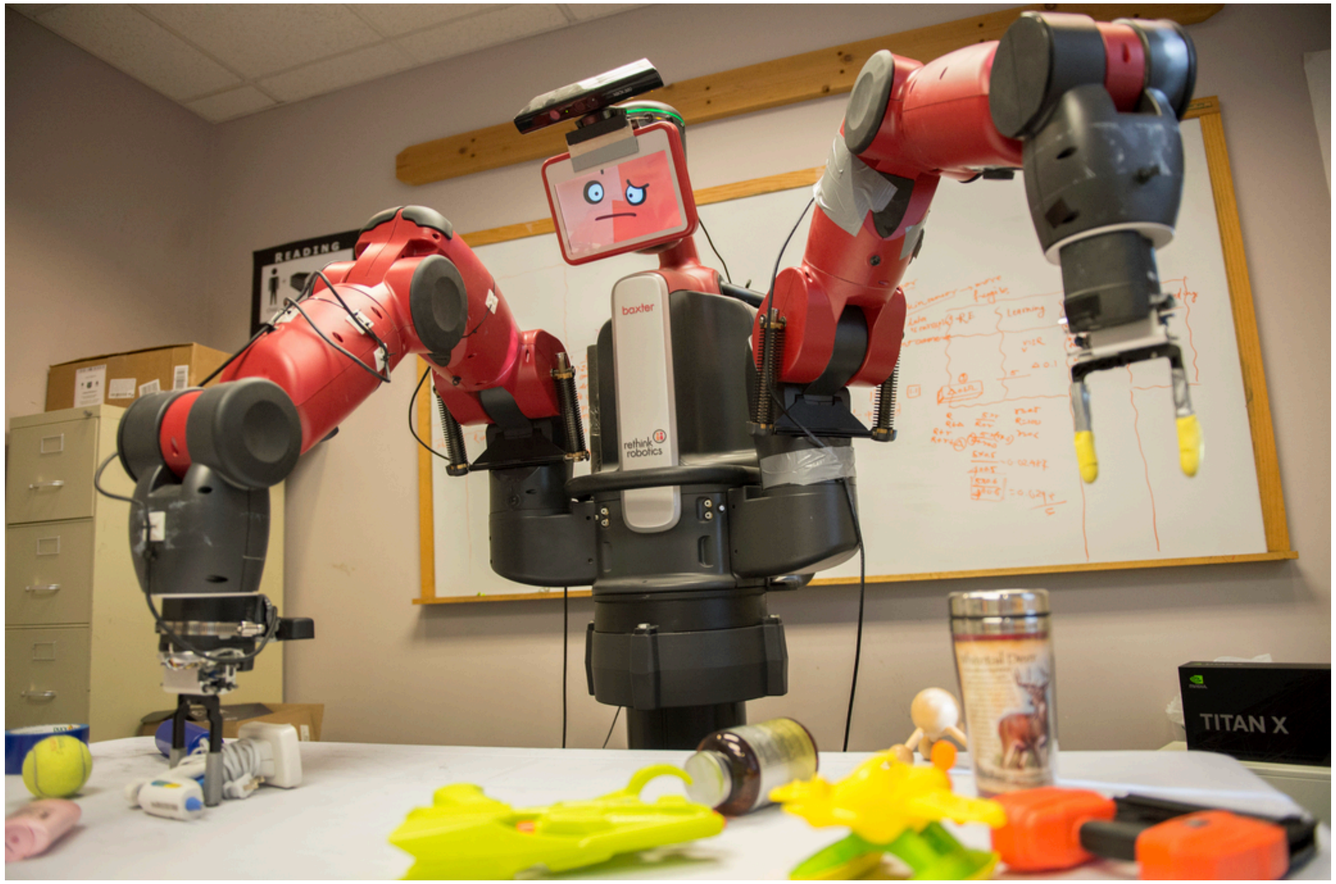
# Example: Particle filters / HMMs

- Particle filters still widely used for robot localization
- Hidden Markov Models are sequence models that preceded LMMs and were often used for part-of-speech tagging and language generation
- HMMs also closely related to POMDPs, which are now often solved by RNNs and Transformers



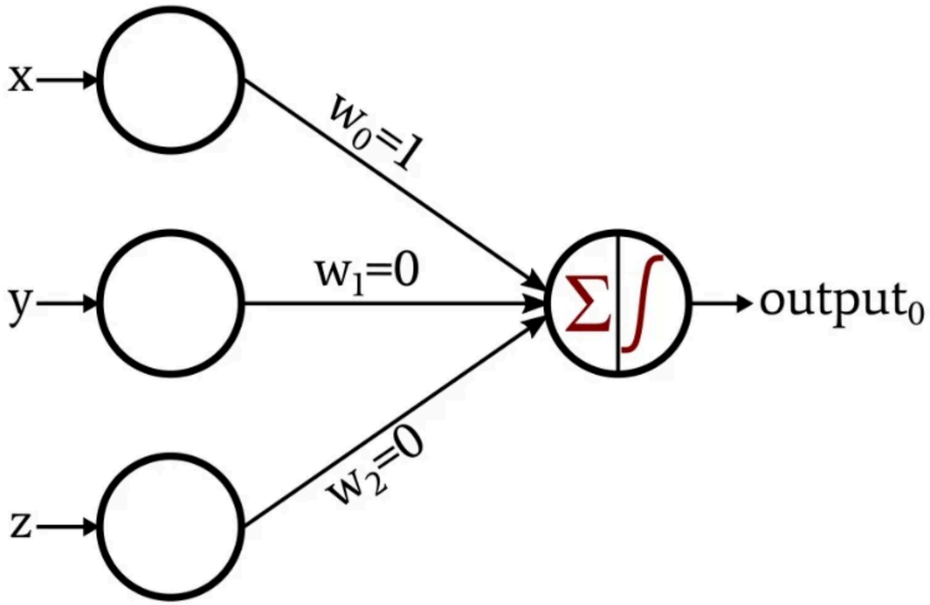
# Example: Value of information

- Active learning



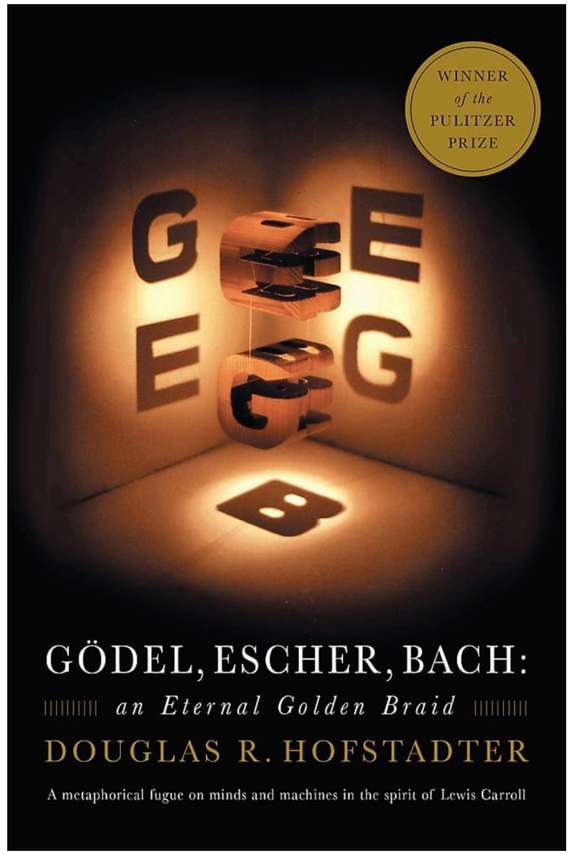
# Example: Perceptrons and kernel methods

- Perceptrons are a highly simplified model of a single neuron. Building block of neural networks.
- Kernel methods closely related to the attention mechanism in transformers
- Neural tangent kernel helps researchers understand the training dynamics of neural networks



# Philosophical questions

- AI is one of the great intellectual adventures of the 20th and 21st centuries.
  - What is intelligence?
  - What is a mind?
  - What is consciousness?
  - Can a computer have these?
  - Can we build these? And how?



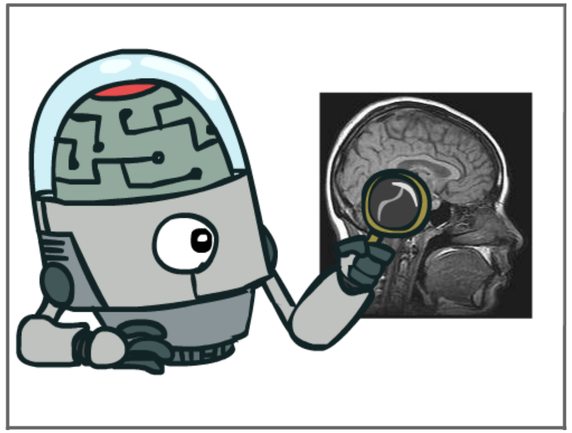
# What is AI?

The science of making machines that:

# What is AI?

The science of making machines that:

Think like people

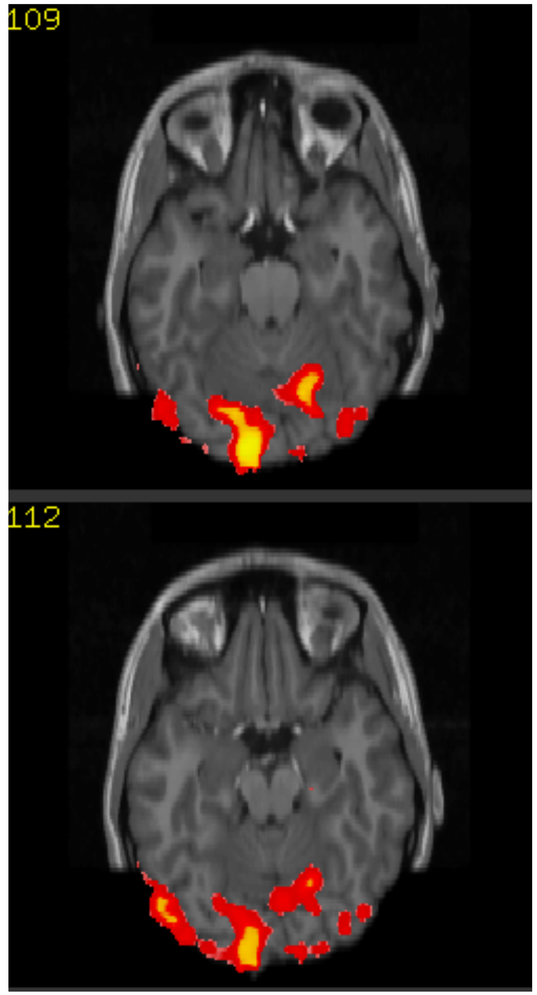


# Thinking Like Humans?

- The cognitive science approach:
  - 1960s ``cognitive revolution'': information-processing psychology replaced prevailing orthodoxy of behaviorism (reflexive behaviors, classical conditioning, etc.)

- Scientific theories of internal activities of the brain
  - What level of abstraction? “Knowledge'' or “circuits”?
  - **Cognitive science:** Predicting and testing behavior of human subjects (top-down)
  - **Cognitive neuroscience:** Direct identification from neurological data (bottom-up)
  - Both approaches now distinct from AI
  - *The available theories do not yet come close to explaining human intelligence and thinking*

- Even if possible, what is the utility of having AI think like humans?

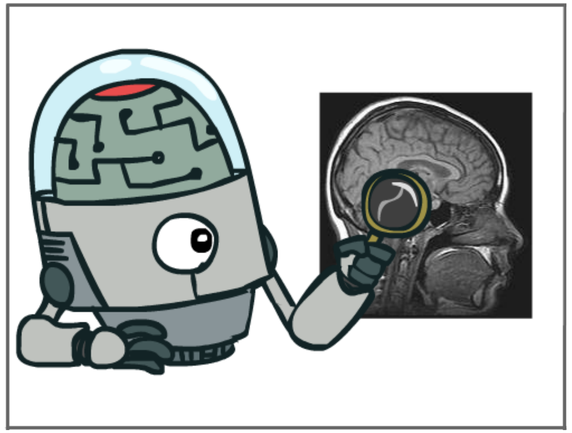



Images from Oxford fMRI center

# What is AI?

The science of making machines that:

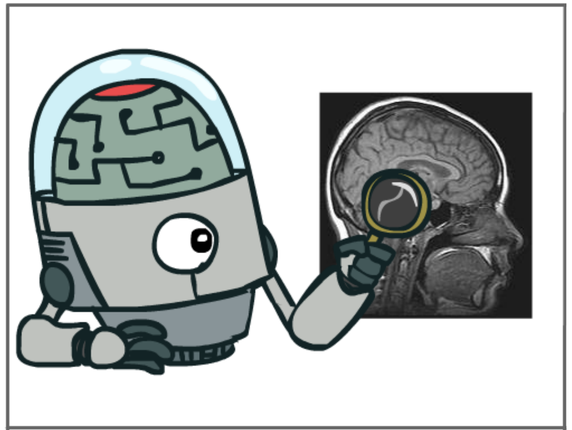
Think like people



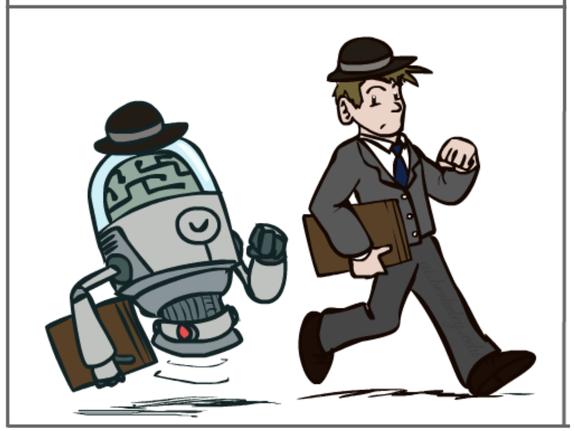
# What is AI?

The science of making machines that:

Think like people

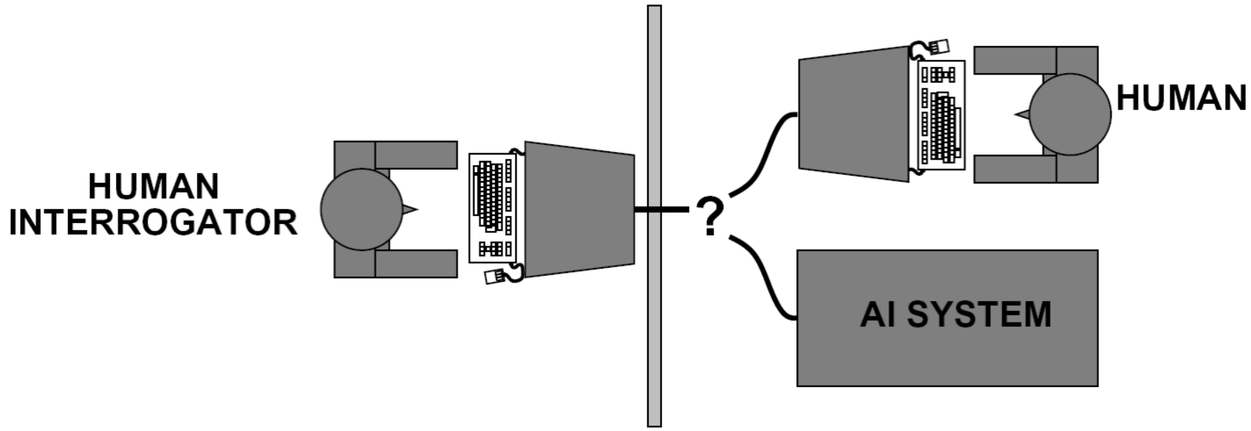


Act like people



# Acting Like Humans?

- Turing (1950) “Computing machinery and intelligence”
  - “Can machines think?” → “Can machines behave intelligently?”
  - Operational test for intelligent behavior: the *Imitation Game*

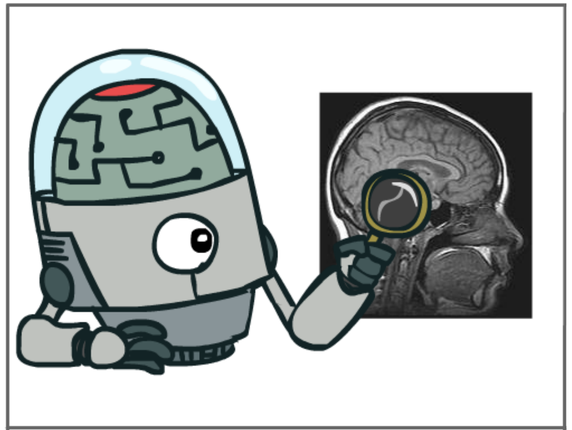



  - Predicted by 2000, a 30% chance of fooling a lay person for 5 minutes
  - Anticipated all major arguments against AI in following 50 years
  - Suggested major components of AI: knowledge, reasoning, language understanding, learning
- Problem: Does the Turing test really measure what we want?

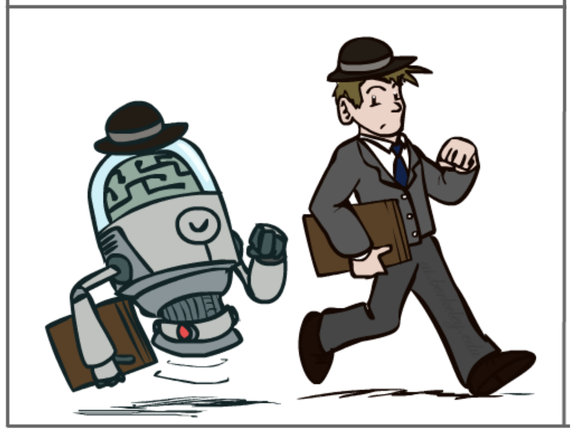
# What is AI?

The science of making machines that:

Think like people



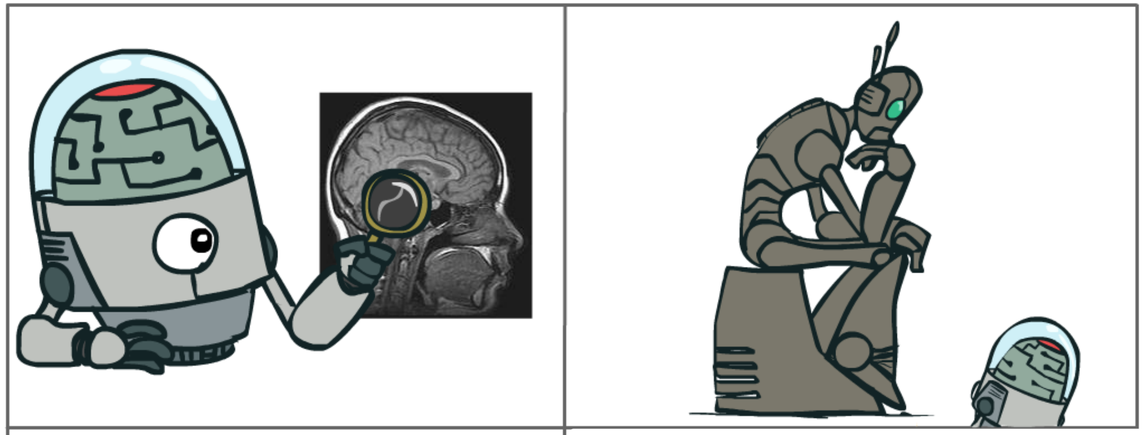
Act like people



# What is AI?

The science of making machines that:

Think like people



Think rationally

Act like people

# Thinking Rationally?

- The “Laws of Thought” approach
  - What does it mean to “think rationally”?
  - Normative / prescriptive rather than descriptive

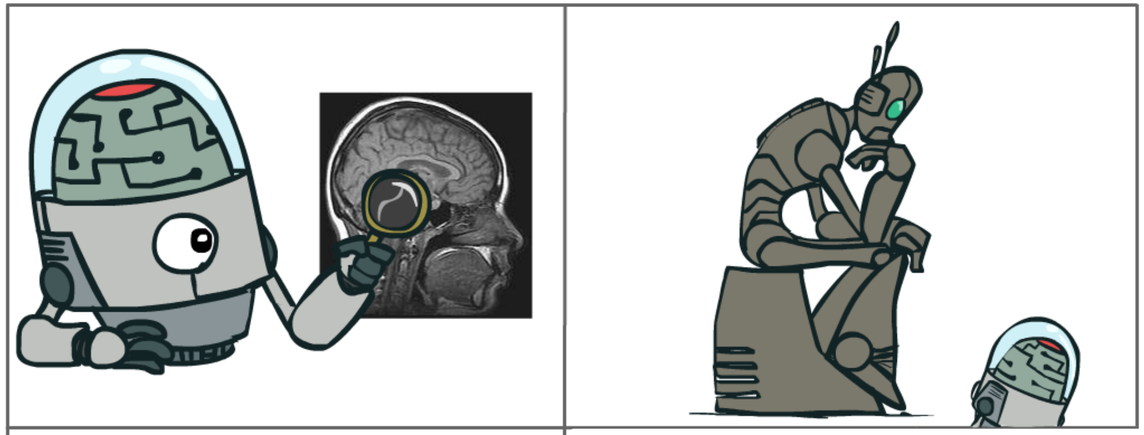
- Logicist tradition:
  - Logic: notation and rules of derivation for thoughts
  - Aristotle: what are correct arguments/thought processes?
  - Direct line through mathematics, philosophy, to modern AI

- Problems:
  - Not all intelligent behavior is mediated by logical deliberation
  - What is the purpose of thinking?  What thoughts should I (bother to) have?
  - Logical systems tend to do the wrong thing in the presence of **uncertainty**
  - Why should we care about thought at all, when **action** is what matters?

# What is AI?

The science of making machines that:

Think like people



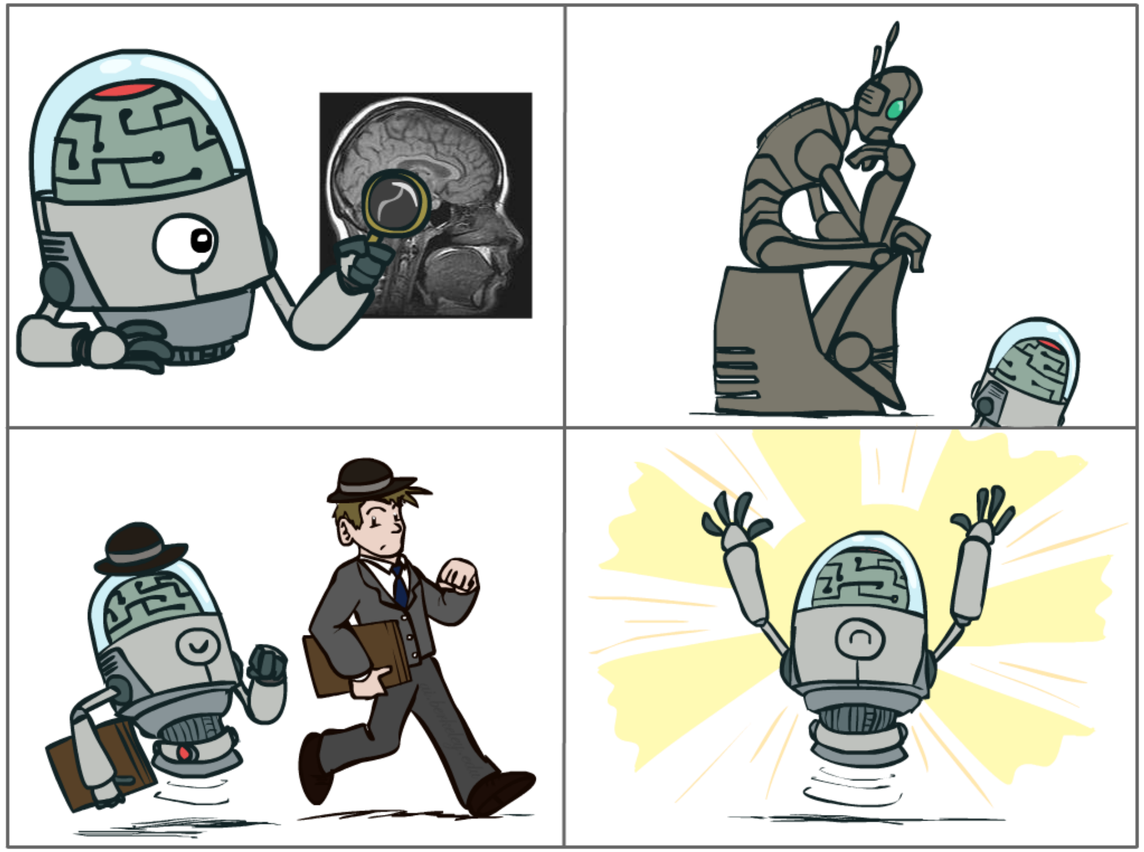
Think rationally

Act like people

# What is AI?

The science of making machines that:

Think like people



Think rationally

Act like people

Act rationally

# Acting Rationally

- Rational behavior: doing the “right thing”
  - The right thing: that which is expected to maximize goal achievement, given the available information
  - Doesn't necessarily involve thinking, e.g., blinking
  - Thinking can be in the service of rational action
  - Entirely dependent on goals!
  - Irrational ≠ insane, irrationality is sub-optimal action
  - Rational ≠ successful

- Our focus here: rational agents
  - Systems which make the best possible decisions given goals, evidence, and constraints
  - In the real world, usually lots of uncertainty
    - … and lots of complexity
  - Usually, we’re just approximating rationality

# Rational Decisions

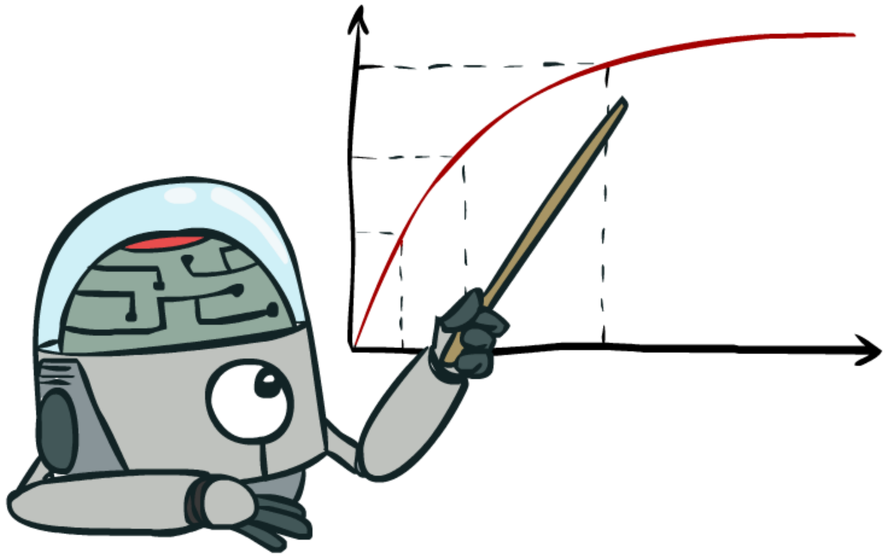
We'll use the term **rational** in a very specific, technical way:

- 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**

A better title for this course would be:

**Computational Rationality**

# Maximize Your Expected Utility



# Course Topics

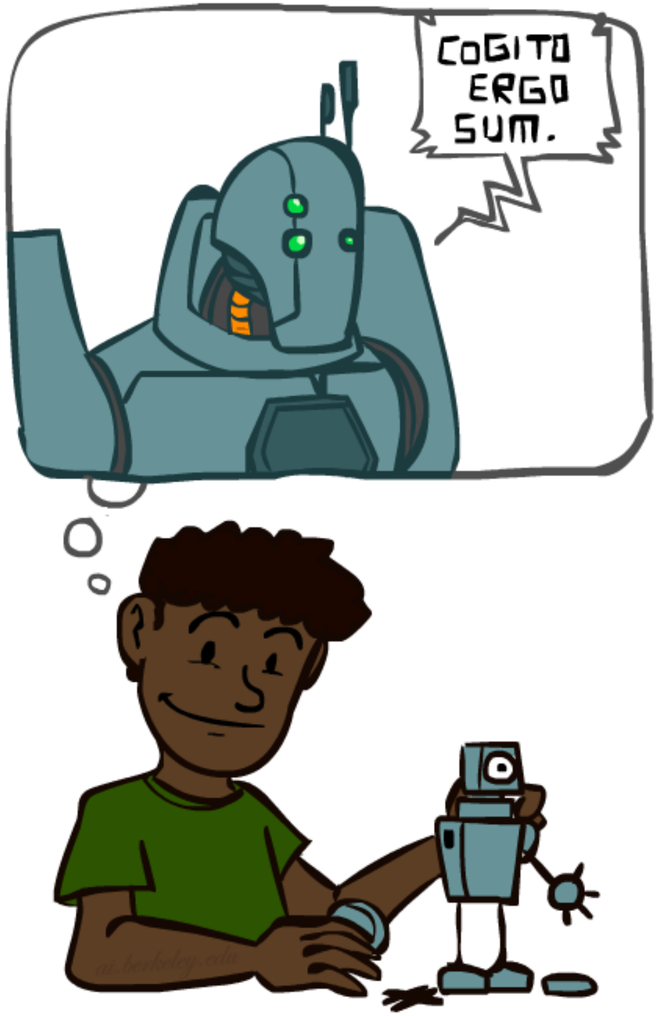
- Part I: Making Decisions
  - Fast search / planning
  - Constraint satisfaction
  - Adversarial and uncertain search
  - MDPs and Reinforcement learning

- Part II: Reasoning under Uncertainty
  - Bayes nets
  - Decision theory and value of information
  - Statistical Machine learning

- Throughout: Applications, Ethics, and Societal impacts

# 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
- 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?



# 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 **performance measure, environment, actions, and sensing** dictate techniques for selecting rational actions
- **By then end of the course** you should understand:
  - General AI techniques for a variety of problem types
  - How to recognize when and how a new problem can be solved with an existing technique