1 What is AI?

- *Computational models of human behavior*: programs that behave (externally) like humans. Turing test. More properly cognitive science.
- *Computational models of human “thought processes”*: programs that operate (internally) the way humans do. More properly computational neuroscience.
- *Computational systems that behave intelligently*: what does that mean?
- *Computational systems that behave rationally*: Our emphasis; more later.
- *AI applications*. Typically built (or possibly to analyze post-hoc) as rational systems.

2 Agents

Focus on systems with ongoing interaction with an external environment.

Model the external environment in terms of

- states
- actions
- observations (percepts)

Models are almost all discrete-time (could use systems of differential equations, but AI folks don’t have much to contribute there).

Model the designer’s goals for the agent using a performance measure on system states or observation streams. These goals may be made explicit inside the agent, or may not, depending on implementation strategy.
3  Rationality

A rational agent takes actions it believes will achieve its goals

- Assume I don’t like to get wet, so I bring an umbrella. Is that rational?
- Rationality is not omniscience: Assume the most recent forecast is for rain, but I did not listen to it and didn’t bring my umbrella. Is that rational?
- Rationality is not success: There is no forecast for rain, but I bring my umbrella anyway and use it to fend off a rabid dog. Is that rational?
- Might be too computationally hard to compute best action, subject to belief and goals
- Limited rationality: choose best actions, subject to your own computational constraints
- Design problem: find mapping of $O^* \rightarrow A$ that maximizes utility of resulting state sequence subject to computational constraints
- This might be too hard a problem for designers to solve.

4  Modeling the Environment

World dynamics: deterministic, nondeterministic, probabilistic
Observations: perfect, nondeterministic, probabilistic

5  Memory

With perfect observations, we need a mapping from observations to actions. Can take long-term reward into account, but doesn’t need internal state (memory).

As soon as observations are not perfect, we need (or at least can benefit from) memory. Agents are mappings from strings of observations into actions.

Belief state is whatever the agent remembers from time step to time step about the environment. Draw figure.

What if:

- initial “belief” is a set of states
- observations are sets of states
- world dynamics are non-deterministic (map a current state to a set of possible next states)
Then belief states can be represented as sets of world states. Consider a problem where a robot perceives only locally (but with no noise) the configuration of walls around it. It starts out not knowing where it is. Whenever it tries to move forward, it moves 0, 1, or 2 squares. A state consists of the robot’s position and orientation.

If the agent has current belief $b$, how should we update it based on taking action $a$ and receiving observation $o$?

Let $n(s, a)$ be the set of possible states that could result from taking action $a$ in state $s$. Then after taking action $a$, our belief is

$$b' = \bigcup_{s \in b} n(s, a).$$

Now, we get observation $o$, which is also a set of possible states. Our new belief state $b''$ is simply $b' \cap o$.

What happens if we travel down a long featureless corridor? What happens if we see a unique doorway configuration? We’ll revisit this question, with probabilities, when we consider POMDPs.

6 System Design Choices

Representation of observations, beliefs: atomic, propositional, first-order

We just talked about a belief state representation as sets of atomic states. But often that will be highly inefficient. Logical languages give us methods for writing down compact descriptions for large sets of states.

For our robot, clear-left might be a logical assertion that describes all of the states of the world in which there is an opening to the left of the robot. Then clear-left $\land$ clear-front could stand for the set of locations in which there are openings both in front of and to the left of the robot.

Tabular v. first-principles

How should we store the functions that map beliefs into new beliefs, or those that map beliefs into actions? We could do it with a gigantic table. That would be very efficient to execute, but huge.

Sometimes a clever human can see how to write an efficient version of a function. An example might be a controller to make a robot drive down the middle of the hallway. This is the strategy followed by the “reactive” or “procedural” programming style.

“Can AI do for intelligent behavior what Newton did for square roots?” - R and N

Some other behaviors, like deciding which way to turn at a given intersection, given the desired goal, seem to complicated to store that way. Instead we “plan” a route. But the map and the search routine together constitute an implementation of the policy function, it’s just somewhat indirect. We’ll call that a “first-principles,” “deliberative,” or “declarative” implementation. There is a general-purpose reasoning engine, together with explicit,
declarative, information about the agent’s current state of knowledge and goals, as well as the nature of the environment.

In general, reactive implementations are more time efficient. “Deliberative” implementations are typically slower, but often much easier for a human to come up with.

**Representing uncertainty** We’ll explore two representations: sets of states and probability distributions over states. But you could imagine others, like: the top 5 most likely worlds, for example.

### 7 AI Keywords

We haven’t used a lot of the standard terms of AI so far. Here’s a way to understand them within this framework.

- **inference**: manipulating logical descriptions of world states to derive explicit statements about particular aspects of the world. Example: computing distance to nearest battery charger, based on location information, map, etc.

- **planning**: inference directed toward action selection; Example: deriving what action is best to take from current state information, model of world dynamics, agent goals.

- **learning**: What if you don’t know very much about the dynamics of the world you’re actually in? You can build all that uncertainty into the world model. The “states” of the world for a learning agent include all the ways that the world could be. As the agent learns, it narrows down its hypotheses about the world and is therefore able to choose actions more effectively.

### 8 Summary

There is no silver bullet. No single true best algorithm or model or representation. We probably need to use a little bit of everything.

Current modern AI gives us a number of tools for this tool-box. Given a real problem, decide:

- How to model environment in terms of states, actions, goals, observations

- Specify what’s known about the environment

- Choose implementation strategy for agent

- Implement and debug ...
Our strategy will be to investigate different representational and algorithmic choices and to understand in which kinds of worlds they work well. In general, the problems we’re trying to solve are impossible or, at best, wildly intractable. So we have to get some leverage by either:

- assuming the world is simple enough that it can be solved optimally
- performing only approximately optimally

In this course, we’ll study:

- Logical representation, inference, and action selection
- Probabilistic representation, inference, and action selection
- Learning (but not basic supervised learning)

Lots of important stuff we’re not going to cover in any detail: natural language, vision, multi-agent systems. Just because we don’t have time, not because they’re not crucial.

9 Exercises

Read chapters 1 and 2 of AIMA. Think through some of these problems: 2.3, 2.4, and maybe 2.12.