January 17, 2024 Notes

0.1 Typical Agent Architectures

- Simple reflex agent: uses condition-action rules

- Model-based reflex agent: keeps track of the unobserved parts of the environment by maintaining internal state:

  - "how the world works": state transition model
  - how percepts and environment is related: sensor model

- Goal-based reflex agent: maintains the models of the world and goals to select decisions (that lead to goal)

- Utility-based reflex agent: maintains the model of the world and utility function to select PREFERRED decisions (that lead to the best expected utility: avg (EU * p))

0.2 State and Transition Representations

- Atomic: state representation has NO internal structure

- Factored: state representation includes fixed attributes (which can have values)

- Structured: state representation includes objects and their relationships

0.3 Problem-Solving / Planning Agent

- Context / Problem:

  - correct action is NOT immediately obvious
  - a plan (a sequence of actions leading to a goal) may be necessary

- Solution / Agent:

  - come up with a computational process that will search for that plan

- Planning Agent:

  - uses factored or structured representations of states
  - uses searching algorithms

0.4   Defining Search Problem

- Define a set of possible states: State Space

- Specify Initial State

- Specify Goal State(s) (there can be multiple)

- Define a FINITE set of possible Actions for EACH state in the State Space

- Come up with a TRANSITION model which describes what each action does

- Specify the Action COST Function (a function that gives the cost of applying action a to state s)

0.5   Measuring Searching Performance

Search algorithms can be evaluated in four ways:

- Completeness: Is the algorithm guaranteed to find a solution when there is one, and to correctly report failure when there is not?

- Cost optimality: Does it find a solution with the lowest path cost of all solutions?

- Time complexity: How long goes it take to find a solution? (in seconds, actions, states, etc.)

- Space complexity: How much memory is needed to perform the search?

0.6   Informed Search and Heuristics

Informed search relies on domain-specific knowledge / hints that help locate the goal state.

$$h(n) = h(\text{State } n)$$
$$h(n) = n(\text{relevant information about State } n)$$

0.7   Romanian Roadtrip: Heuristics h(n)

For this particular problem, the heuristic function $h(n)$ is defined by a straight line (Euclidean) distance between two states (cities). As the crow flies in other words.