# CS 581 Notes

## 0.1 Agent

An agent is just something that acts (from the Latin agere, to do).

Of course, we would prefer "acting" to be:

- autonomous
- situated in some environment (that could be really complex)
- adaptive
- create and goal-oriented

## 0.2 Rational Agent

A rational agent is one that acts to achieve the best outcome, or when there is uncertainty, the best expected outcome[1].

## 0.3 AI: Constructing Agents

You can say that: AI is focused on the study and construction of agents that do the right thing.

## 0.4 Percepts and Percept Sequences

Percept – content/information that agent's sensors are perceiving / capturing currently

Percept Sequence – a complete history of everything that agent has ever perceived

- any practical issues that you can see here?
- what can a percept sequence be used for?

## 0.5 Agent Function/Program

Specifying an action choice for every possible percept sequence would define an agent

- Action <-> percept sequence mapping IS the agent function.
- Agent function describes agent behavior.

[1] no worries, we will make it a little less vague soon

- Agent function is an abstract concept.
- Agent program implements agent function.

## 0.6 Actions Have Consequences

An agent can act upon its environment, but how do we know if the end result is "right"? After all, actions have consequences: either good or bad. Recall that agent actions change environment state! If state changes are desirable, and agent performs well. Performance measure evaluates state changes.

## 0.7 Rationality

Rational decisions at the moment depend on:

- The performance measure that defines success criteria
- The agent's prior knowledge of the environment
- The actions that the agent can perform
- The agent's percept sequence so far

## 0.8 Rational Agent

For each possible percept sequence, a rational agent should select an action that is expected to maximize

## 0.9 Rationality in Reality

- An omniscient agent will ALWAYS know the final outcome of its action. Impossible in reality. That would be perfection.
- Rationality maximizes what is EXPECTED to happen.
- Perfection maximizes what WILL happen.
- Performance can be improved by information gathering and learning.

## 0.10 Designing the Agent for the Task

### 0.10.1 Analyze the Problem

**Task Environment — PEAS**

In order to start the agent design process, we need to specify/define:

- The Performance measure
- The Environment in which the agent will operate
- The Actuators that the agent will use to affect the environment
- The Sensors

**Task Environment Properties**

Key dimensions by which task environments can be categorized:

- Fully vs partially observable (can be unobservable too)
- Single agent vs multi-agent
  - multi-agent: competitive vs. co-operative
- Deterministic vs. non-deterministic (stochastic)
- Episodic vs. sequential
  - Sequential requires planning ahead
- Static vs. dynamic
- Discrete vs. continuous
- Known vs. unknown (to the agent)

### 0.10.2 Select Agent Architecture

$$\textbf{Agent} = \textbf{Architecture} + \textbf{Program}$$

**Typical Agent Architectures**

- Simple reflex agent.
- Model-based reflex agent.
- Goal-based reflex agent.
- Utility-based reflex agent.

### 0.10.3 Select Internal Representations

## 0.11 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))

## 0.12 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

## 0.13 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

## 0.14 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 $s$)

## 0.15 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?

## 0.16 Informed Search and Heuristics

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

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

## 0.17 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.

## 0.18 A* Algorithm: Evaluations Function

Calculate/obtain:

$$f(n) = g(\text{State}_n) + h(\text{State}_n)$$