Rational Agents: Can Computers Think? (How do Computers Think?)

Tas: Elena Filatova
Michel Galley
Sowmya Viswanath

Reading: Chapter 2 (today)
Chapter 3 (Thursday)
Four intelligent agents

- Robotic vacuum cleaner (Roomba)
- Robotic soccer player
- A system that can solve cryptograms
- A system that can play chess
Cryptograms

QFL HCVPS

PX V ANSWLCEZK NCJVS; PQ XQVCQX
QFL BPSZQL RNZ JLQ ZT PS QFL
BNCSPSJ VSW WNLX SNQ XQNT
ZSQPK RNZ JLQ QN QFL NEEPGL
CNHLCCQ ECNXXQ
Cryptograms

QFL HCVPS
THE BRAIN

PX V ANSWLCEZK NCJVJS; PQ XQVCQX
IS A WONDERFUL ORGAN; IT STARTS

QFL BPSZQL RNZ JLQ ZT PS QFL BNCSPSJ
THE MINUTE YOU GET UP IN THE MORNING

VSW WNLX SNQ XQNT ZSQPK RNZ JLQ QN
AND DOES NOT STOP UNTIL YOU GET TO

QFL NEEPGL
THE OFFICE

CNHLCQ ECNXQ
ROBERT FROST
Components of a rational agent

- A performance measure that defines success
- The agent’s knowledge of environment
- The actions the agent can perform
- The agent’s percept sequence to date
  - What has the agent determined from the environment so far?
A performance measure that defines success

- Roomba:
- Soccer:
- Cryptogram:
- Chess:

The agent’s knowledge of environment

- Roomba:
- Soccer:
- Cryptogram:
- Chess:
The actions the agent can perform

- Roomba:
- Soccer:
- Cryptogram:
- Chess:

The agent’s precept sequence to date

- What has the agent determined from the environment so far?
  - Roomba:
  - Soccer:
  - Cryptogram:
  - Chess:
Definition of a rational agent

- For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever knowledge the agent has built in.
Properties of task environment

- Fully observable
- Deterministic
- Episodic
- Static
- Discrete
- Single agent

- Partially observable
- Stochastic
- Sequential
- Dynamic
- Continuous
- Multiagent
Roomba

- Fully observable
- Deterministic
- Episodic
- Static
- Discrete
- Single agent

- Partially observable
- Stochastic
- Sequential
- Dynamic
- Continuous
- Multiagent
Soccer

- Fully observable
- Deterministic
- Episodic
- Static
- Discrete
- Single agent

- Partially observable
- Stochastic
- Sequential
- Dynamic
- Continuous
- Multiagent
Cryptograms

- Fully observable
- Deterministic
- Episodic
- Static
- Discrete
- Single agent

- Partially observable
- Stochastic
- Sequential
- Dynamic
- Continuous
- Multiagent
Chess

- Fully observable
- Deterministic
- Episodic
- Static
- Discrete
- Single agent

- Partially observable
- Stochastic
- Sequential
- Dynamic
- Continuous
- Multiagent
Simple Reflex Agent

- Can select actions based on current percept
- Condition-action rules

Function `Simple-Reflex-Agent` (percept) returns an action. *Static*: rules, a set of condition-action rules

- `State` ← Interpret-input (percept)
- `Rule` ← Rule-Match (state, rules)
- `Action` ← Rule-Action(rule)

Return *action*
Simple Reflex Agent
Model-based Reflex Agents

- Agent maintains internal state
  - E.g., part of the world it can’t see now
- Agent maintains a model of the world

- Function Reflex-Agent-with-state (percept) returns an action.
  Static: rules, a set of condition-action rules; state, a description of the current world state; action, the most recent action, initially none
  - State $\leftarrow$ Update-state (state, action, percept)
  - Rule $\leftarrow$ Rule-Match (state, rules)
  - Action $\leftarrow$ Rule-Action(rule)
- Return action
Model-based Reflex Agents
Goal-based Agents

- Agents that work towards a goal
  - Select the action that more likely achieve the goal
- Sometimes an action directly achieves a goal; sometimes a series of actions are required
Goal-based Agents
Utility-based Agents

- How much better is one state than another?
  - Utility function generates a number for a state indicating how good it is

- Situations in which utility is needed:
  - Conflicting goals
  - Several possible goals
Utility-based agents
Problem solving as search

- Goal formulation
- Problem formulation
- Actions
- States
Formulating Problems as Search

Given an **initial state** and a **goal**, find the sequence of **actions** leading through a sequence of **states** to the final **goal state**.

Terms:

- **Successor function**: given action and state, returns \{action, successors\}
- **State space**: the set of all states reachable from the initial state
- **Path**: a sequence of states connected by actions
- **Goal test**: is a given state the goal state?
- **Path cost**: function assigning a numeric cost to each path
- **Solution**: a path from initial state to goal state
Formulating cryptograms as search

- Initial state
  
  QFL HCVPS
  
  PX V ANSWLCEZK NCJVS; PQ XQVCQX QFL BPSZQL RNZ JLQ ZT PS QFL BNCSPSJ VSW WNLX SNQ XQNT ZSQPK RNZ JLQ QN QFL NEEPGL CNHLCQ ECNXQ

- Goal state
  
  QFL HCVPS
  
  THE BRAIN

  PX V ANSWLCEZK NCJVS; PQ XQVCQX QFL BPSZQL RNZ JLQ ZT PS QFL IS A WONDERFUL ORGAN; IT STARTS THE MINUTE YOU GET UP IN THE MORNING AND DOES NOT STOP UNTIL YOU GET TO THE OFFICE CNHLCQ ECNXQ

  ROBERT FROST
Cryptograms

- Action?
- Successor states?
- State space?
- Path cost?