COMP3702
Artificial Intelligence

Tutorial 1: Introduction

Semester 2, 2022

The University of Queensland
School of Information Technology and Electrical Engineering

Dimensions of complexity in agent design

From P&M Ch 1.5:

Dimension	Values
Modularity:	flat, modular, hierarchical
Planning horizon:	non-planning, finite stage, indefinite stage, infinite stage
Representation:	states, features, relations
Computational limits:	perfect rationality, bounded rationality
Learning:	knowledge is given, knowledge is learned
Sensing uncertainty:	fully observable, partially observable
Effect uncertainty:	deterministic, stochastic
Preference:	goals, complex preferences
Number of agents:	single agent, multiple agents
Interaction:	offline, online

Agent Design

To solve an **agent design problem**, the following components are required:

- **Action Space** (A): The set of all possible actions the agent can perform (sometimes called the *action set* in the discrete case). An action is denoted $a \in A$.
- **Percept Space** (P): The set of all possible things the agent can perceive.
- **State Space** (S): The set of all possible configurations of the world the agent is operating in (sometimes called the *set* of states in discrete state systems). A state is denoted $s \in S$.
- **World Dynamics/Transition Function** ($T : S \times A \to S'$): A function that specifies how the world changes when the agent performs actions in it; a system model. We sometimes write $T(s, a) = s'$.
- **Perception Function** ($Z : S \to P$): A function that maps a world state to a perception.
- **Utility Function** ($U : S \to \mathbb{R}$): A function that maps a state (or a sequence of states) to a real number, indicating how desirable it is for the agent to occupy that state/sequence of states. We sometimes write $U(s) =$ some cost or reward.

Exercise 1.1

Exercise 1.1 - Tic-Tac-Toe

- Action Space - Place element in position $\{i \mid i \epsilon Z,\ i \epsilon [0,9)\}$
- Percept Space - All placed elements
- State Space - All combinations of elements placed
- Transition Function - Given a state S and performing an action A results in a new state S'. e.g. S(000000000) x A(X,(1,1)) = S'(0000X0000)
- Utility Function
    - Can be 0 - random
    - Can be the number of ways we can still win

Exercise 1.2

Exercise 1.2 - Navigation App

- Action Space - Given a set of all roads, select road and tell user direction
- Percept Space - Map data, not traffic or pedestrians
- State Space - (x,y) coordinates where $x, y \epsilon R$
- Transition Function - $S(x, y)$ x $Direction(\overrightarrow{x'}, \overrightarrow{y'}) = S'(x', y')$
- Utility Function - Each state (x,y) euclidean distance from destination, so utility function is $-$(distance to destination). This is informed search, more in coming tutorials.

Exercise 1.2 - Navigation App

- Agent can be discrete or continuous
    - Intersections - Discrete
    - Real world coordinates - Continuous
- Agent is non-deterministic, the agent can only 'tell' the driver what to do, there is no guarantee the driver will do it :P
- GPS gives fully observable map data - traffic and pedestrians are not observable
- Agent is dynamic - There are external processes operating on the agent

Exercise 1.3

Exercise 1.3 - Web Crawler

- Action Space - Find links and crawl
- Percept Space - Text within the set of all links collected
- State Space - All the links on the internet
- Transition Function - Given a new link, select it, finding new links and adding them to collection
- Utility Function - $+1$ for new link found

Exercise 1.4

Exercise 1.4 - Poker Bot

- Action Space - Check, Call, Raise, Fold
- Percept Space - The agents 2 cards and the cards shown in each stage
- State Space - It can get complicated, try all the following
  - All combinations of cards for each player including shown cards
  - All combinations of current and previous bets from all players
  - All combinations of total chips for each player and in pot
- Transition Function - Given current state S of (cards,chips,bets..etc), perform action (Check, Call, Raise, Fold), to arrive to a new state S', where everyone folds and you win all the chips.
- Utility Function
  - +X for winning where X is the total pot
  - +5 for an opponent that folds
  - +10 for hitting a card that improves your hand