COMP90054 – Week 2 tutorial

Prepared by Zijie Xu. Last updated: 21 March 2024

Admin and Communication

Contact

- Tutor name: Zijie Xu (Jerry)
- Tutorial time
  - Thursday 1 pm – 109, 100 Leicester St
  - Friday 9 am – WG05, John Medley building
- E-mail: zijie.xu2@unimelb.edu.au

Lecturers

- Dr Nir Lipovetzky, Dr Joseph West, Dr Sarita Rosentock

Discussion forum

- Preferably, post your questions on the Ed discussion forum

Some prior knowledge

- Algorithms, data structures (e.g. Dijkstra, dynamic programming)
- Basic set theory and Propositional Logic (e.g. $\in \quad \subseteq \quad \wedge \quad \vee \quad \cap \quad \cup$)
- Basic probability (e.g. conditional probability)
- Python (with some knowledge in object-oriented programming)

Prep work before each tutorial

- Review/Catch-up with last week's lecture material
- Watch the amazing pre-tute prep videos prepared by Guang
- Save a copy of Tutorial Colab Notebook and play with it

Intro to AI and Planning

Rationally acting agents

- Knowledge & Percepts & Performance measure -> Action
- Maximise expected performance given its percept and knowledge
- **Problem -> Solver -> Solution**

Classical planning

Problem	->	Solver	->	Solution
Model	->	Planner	->	Action sequence

Planning is a **model-based approach** to autonomous behaviour:

- We use **model** to specify a **problem** P
- The system/environment can be in one of many **states**
- Task of planning: Find **action sequence** to drive **initial state** into **goal state**
- Submit model to **planner** so it can come up with a **plan** (action sequence)

State-space model

$$\mathcal{S}(P) = \langle S, s_0, S_G, A, f, c \rangle$$

S	State space S, finite and discrete
s_0	A **known** initial state s_0 (c.f. belief of initial state)
S_G	A set of goal states $S_G \subseteq S$
A	A set of actions, with $A(s) \subseteq A$ for each $s \in S$
f	A **deterministic** transition function $f: (a, s) \to s'$ for $a \in A(s)$
c	Positive **action cost** functions $c(a, s)$

Blind search algorithms

Why graph search?

- Model -> **Planner** -> Action sequence (Plan)
- We want to solve problem P by using **graph search algorithms** over the graph associate with $\mathcal{S}(P)$

Blind Search algorithms

- "Blind" means not requiring any input beyond problem P (c.f. Heuristics/informed search algorithms, coming up soon!)
- Algorithms
    - Breadth-First Search (BrFS)
    - Depth-First Search (DFS)
    - Iterative Deepening (ID)

How to extract plans

By following/reversing steps of the search node expansion in search space

A search node σ contains

- $state(\sigma)$ Associated search state
- $parent(\sigma)$ Pointer to search node from which σ is reached $state(\sigma)$
- $action(\sigma)$ An action leading from $state\big(parent(\sigma)\big)$ to
- $g(\sigma)$ Cost of path from the root node to σ