THE UNIVERSITY OF QUEENSLAND
AUSTRALIA
CREATE CHANGE

COMP3702
Artificial Intelligence

Tutorial 6

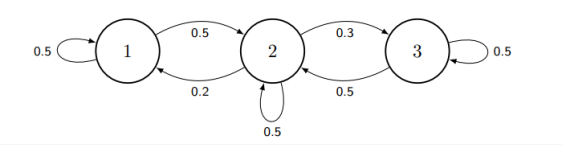
Aryaman Sharma (aryaman.sharma@uq.edu.au)

Semester 2, 2022

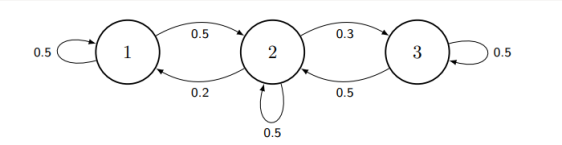
The University of Queensland
School of Information Technology and Electrical Engineering

Markov chains

Markov chains



Markov chains



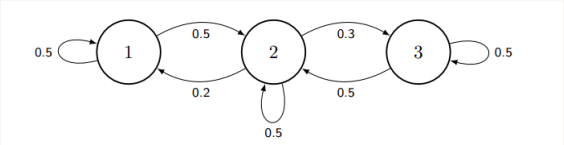
Definition (Markov Chain)

A collection of random variables $X_1, X_2, ...$ is called a **Markov chain** with **State space** E if

$$\mathbb{P}(X_{n+1} = x_{n+1} | X_0 = x_0, ..., X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)$$

for all $x_0, ..., x_{n+1} \in E$ and $n \in N$

Markov chains



Definition (Markov Chain)

A collection of random variables $X_1, X_2, ...$ is called a **Markov chain** with **State space** E if

$$\mathbb{P}(X_{n+1} = x_{n+1} | X_0 = x_0, ..., X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)$$

for all $x_0, ..., x_{n+1} \in E$ and $n \in N$

In essence the transition probability does not depend on history.

Markov chains

- One step transition matrix

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots \\ p_{21} & p_{22} & \cdots \\ \vdots & \vdots & \end{pmatrix}$$

Markov chains

- One step transition matrix

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots \\ p_{21} & p_{22} & \cdots \\ \vdots & \vdots & \end{pmatrix}$$

- Initial state probability $x_0 = [0, 0, 1, ...]$

Markov chains

- One step transition matrix

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots \\ p_{21} & p_{22} & \cdots \\ \vdots & \vdots & \end{pmatrix}$$

- Initial state probability $x_0 = [0, 0, 1, ...]$
- $x_k = x_0 P^k = x_{k-1} P = (x_{k-2} P) P ...$

Markov chains

- One step transition matrix

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots \\ p_{21} & p_{22} & \cdots \\ \vdots & \vdots & \end{pmatrix}$$

- Initial state probability $x_0 = [0, 0, 1, ...]$
- $x_k = x_0 P^k = x_{k-1} P = (x_{k-2} P) P ...$
- **Stationary distribution** is a probability distribution that remains unchanged in a markov chain as time progresses.

$$xP = x$$

Markov chains

■ **Example 6.1 (Stepping Stones)** Suppose there are 5 rocks in a pond on which you can step, see Figure 6.1. The arrows indicate from which stone you can step to another. Suppose that from your current position you choose your next stone with equal probability amongst the possible stone. For example, if, at after a number of steps you are on stone 5, you can step onto stone 4 or stone 6, with probability 1/2 each. Suppose you start from position (stone) 1. Where would you be after 10 steps? What would the probability be that you end up back in position 1 after 10 steps?

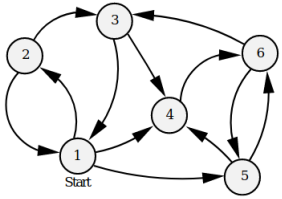
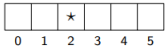


Figure 6.1: Stepping Stones

Exercise 6.1

Exercise 6.1

Exercise 6.1. Restless robot on a Markov chain. Consider the following gridworld:



Imagine you have a restless robot with no AI implemented on it. When you put your robot on the gridworld above, it moves in a random direction, left or right, from one square to an adjacent square with equal probability at each time step. The only time your restless robot stays still is if it tries to move into one of the ends of the gridworld, i.e. left in state 0 or right in state 5.

The motion of your restless robot can be modeled as a Markov chain. This particular Markov chain is called a *simple random walk* on a finite set of integers. For example, consider starting in state 2, as indicated by the star: the probability of moving left, to state 1, is 0.5, and the probability of moving right, to state 2, is also 0.5.

(a) What are the dimensions of this Markov chain's transition matrix?

(b) Construct the transition matrix, P. (Do this in code). Take care with states 0 and 5.

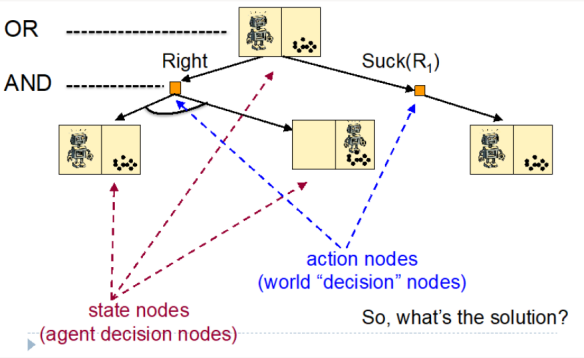
(c) Give the initial state probability vector, x_0 for starting in state 2, and compute the distribution of states of the Markov chain at $t = 1$.

(d) Using matrix multiplication methods, compute the distribution of states of the Markov chain 2 time steps after starting in state 2, x_2, by multiplying again by P. Repeat the multiplication for 4, 10 and 20 time steps. What is happening to the resulting distribution over states?

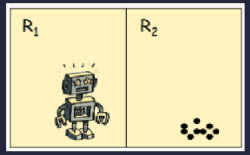
AND-OR Trees

AND-OR Trees

- Used where the action to perform depends on the output of the previous action.
- Situations where a choice is made (e.g. choosing which road to follow at an intersection) are represented by **OR nodes**.
- Random outcomes (e.g. different amounts of time taken to reach the end of a road due to traffic on that road) and represented by **AND nodes**.



Slippery Vacuum Robot



- The state of the robot can be described as follows:
  - The robot's position is either {in R1, in R2}
  - R1 = {clean, dirty}
  - R2 = {clean, dirty}
- The robot can perform the following actions:
  - {Left, Right, Suck(R1), Suck(R2)
- The world is non-deterministic – the robot will sometimes perform the desired movement action
- The initial state of the robot can be described as follows:

$$(Robot\ in\ R_1) \wedge (R_1\ is\ clean) \wedge (R_2\ is\ dirty)$$

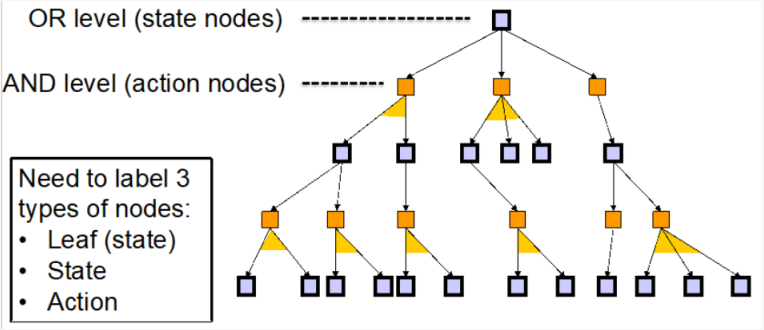
- The goal state is as follows:

$$(R_1\ is\ clean) \wedge (R_2\ is\ clean)$$

AND-OR Trees

- Solution is a sub-tree that:
  - Has a goal node at every leaf
  - Specifies one action at each node of an OR level
  - • Includes every outcome branch at each node of an AND level
-

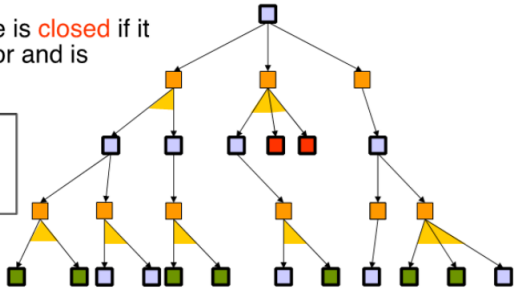
Labelling AND-OR Trees



Labelling AND-OR Trees

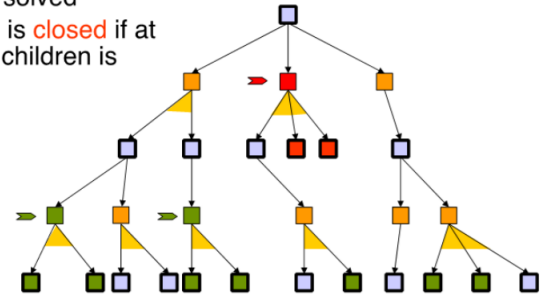
- A leaf state node is solved if it's a goal state
- A leaf state node is closed if it has no successor and is not a goal

Can view:
- Solved as True
- Closed as False



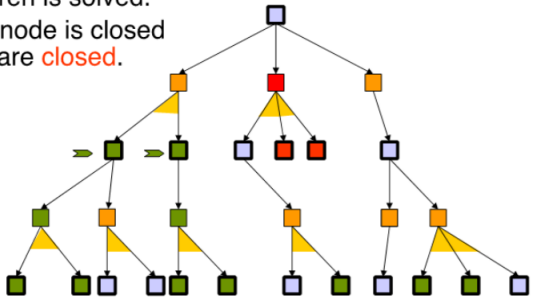
Labelling AND-OR Trees

- An action node is solved if all its children are solved
- An action node is closed if at least one of its children is closed



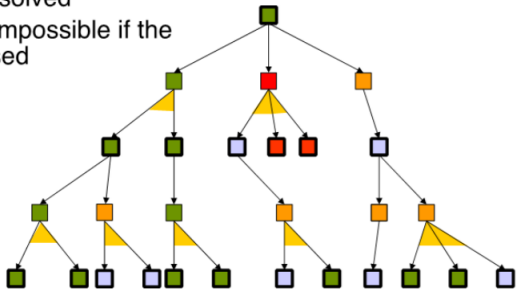
Labelling AND-OR Trees

- A non-leaf state node is solved if one of its children is solved.
- A non-leaf state node is closed if all its children are closed.



Labelling AND-OR Trees

- The problem is solved when the root node is solved
- The problem is impossible if the root node is closed



Labelling AND-OR Trees

Start from a state node (OR level)

- Fringe nodes are state nodes

Use any of the search algorithms we have studied,

- Select a fringe node to expand
- Select an action to use
- Insert the corresponding action node
- Insert all possible outcomes of the action, as the child of the action node
- Backup to (re-)label the ancestor nodes

Cost/reward calculation at AND level:

- Weighted sum (when uncertainty is quantified using probability, expectation)
- Take the maximum cost / minimum reward (conservative)

Exercise 6.2

Exercise 6.2

Exercise 6.2. Suppose your friend is developing a navigation agent for Brisbane metro (the starting and end position is within Brisbane CBD area). His goal is for the agent to recommend a path that can reach the goal within a given time duration. To simplify the problem, time duration is discretized into 15-minutes intervals. To account for traffic uncertainty, the agent uses AND-OR tree, with "solved leaf node" being the goal point reached within the given time duration. As discussed in class, the solution is found whenever the root of the tree can be labelled as solved.

Using this approach, if the agent cannot find a solution, will it still be possible for the agent to move to the goal point within the desired time duration? Please explain your answer. To provide a clear explanation, please first define the agent problem for this navigation agent.