Introduction: Problem Environments & Intelligent Agents

CS3243: Introduction to Artificial Intelligence – Lecture 1

9 January 2023

Contents

Reference: AIMA 4th Edition, Chapter 1-2

Administrative Matters

Teaching Staff

- Lecturer
  - Daren Ler : dler@comp.nus.edu.sg
  - https://www.comp.nus.edu.sg/cs/bio/dler/
- Tutors
  - Amadeus Aristo Winarto : e0559265@u.nus.edu
  - Bharath Shankar : e0550582@u.nus.edu
  - Gu Zhenhao : guzh@nus.edu.sg
  - Sai Sumanth Yalamarty : e0638867@u.nus.edu
  - Wira Azmoon Ahmad : e0014172@u.nus.edu

Topics

1. Introduction: Problem Environments and Intelligence Agents
2. Uninformed Search: Problem-solving Agents and Path Planning
3. Informed Search: Incorporating Domain Knowledge
4. Local Search: Goal Versus Path Search
5. Constraint Satisfaction Problems: Generalising Goal Search
6. Adversarial Search: Playing Games
7. Logical Agents: Knowledge Representation
8. Bayesian Networks: Representations within Uncertainty

Weekly Schedule

- Lectures
  - *Mondays, 1000-1200 hrs*
- Diagnostic Quizzes (Optional but Recommended)
  - Release: *same day as lecture (after lecture)*
  - Submission: *None* (attempt independently – review with peers/tutor)
- Tutorials
  - Begin *Week 3*
  - Release: *After lectures (from Lecture 2 onwards)*
  - Tutorial Assignment Deadlines: *In your tutorial session*

Detailed Schedule: Canvas > CS3243 > About the CS3243 Course > Schedule

Project and Examination Schedule

- Projects
  - Project 1: released *Week 3* ; due *Week 6*
  - Project 2: released *Week 6* ; due *Week 9*
  - Project 3: released *Week 9* ; due *Week 12*
- Midterm Examination
  - *27 February, 1030-1130 hrs* (Week 7 Lecture Slot)
- Final Examination
  - *29 April, 0900-1100 hrs*

Detailed Schedule: Canvas > CS3243 > About the CS3243 Course > Schedule

Expected Learning Flow

1. Attend Lecture
2. Attempt Diagnostic Quiz
3. Complete Tutorial
4. Attend Tutorial / Submit Tutorial Assignment

- Reference Lecture Videos & Textbook as necessary
- Ask on Telegram
- Discuss on Canvas > Discussions

About the Diagnostic Quizzes

- Some positive comments

The weekly diagnostic quiz and tutorial assignments were acceptable workload and helped us to revise content.

I like that it is structured and forces us to keep up with the lectures with weekly quizzes and assignments.

- Some negative comments

The weekly diagnostic quizzes felt a bit too time consuming.

The course had the typical micromanager-y nature of assignments that CS modules tend to have, which I think lecturers should not bother, especially for a 3k module. If a student can't self-manage how much they learn from a module at a 3k level, they kind of deserve to fail.

Assessments

- Tutorial Participation
  - Total 5%
- Tutorial Assignments (9)
  - Total 5%
  - Submission in Tutorials
  - 0.625% each (Best 8)
- Python Projects (3)
  - Total 30%
  - Individual
  - 10% each
- Midterm Examination (1)
  - Total 20%
  - Closed Book + Cheat Sheet (1 × A4 Sheet)
  - In-person + Written
- Final Examination (1)
  - Total 40%
  - Closed Book + Cheat Sheet (1 × A4 Sheet)
  - In-person + Written

About the Projects

- ## Some positive comments

Projects are tough but enjoyable, and we can directly apply what we learnt in class.

The projects have given me exposure to applications of AI, which has made the module more meaningful.

Projects were kind of fun; albeit extremely time consuming.

Challenging but doable projects.

- ## Some negative comments

The 3 projects had very high workload.

The projects are rather time consuming.

Extremely steep learning curve for projects.

The final project should be more strongly about implementing the algorithm, rather than about the evaluation functions.

Plagiarism & Copyright

- Plagiarism
  - Reported to the School for Disciplinary Action

- Copyrights

NUS National University of Singapore

NUS Course Materials: Ethical Behaviour and Respecting Copyright

All course participants (including permitted guest students) who have access to the course materials on LumiNUS or any approved platforms by NUS for delivery of NUS modules are not allowed to re-distribute the contents in any forms to third parties without the explicit consent from the module instructors or authorized NUS officials

NUS National University of Singapore

Examples of Disallowed Things

No Posting on any websites (except for the materials explicitly allowed by your lecturer in the respective module)

No selling of material

No sharing of questions/answers which could lead to cheating/plagiarism

Lecture Protocol

- Post questions anytime
  - Ask questions on Archipelago
- Answers given
  - After the break and at the end of the lecture
- Archipelago
  - Use Voting Board to post/upvote questions

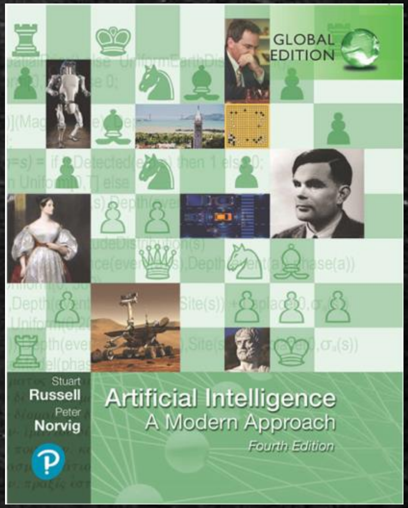
Invitation Link (Use NUS Email --- starts with E)
https://archipelago.rocks/app/resend-invite/50153102954

- Lecture recordings
  - Canvas > CS3243 > Videos/Panopto

Resources & Textbook

- All course material will be on Canvas (Files)
  - https://canvas.nus.edu.sg/courses/38641

- Textbook
  - Artificial Intelligence: A Modern Approach (4th Edition)
  - IBSN 9780134610993
  - Library: https://linc.nus.edu.sg/search~S16?/i=9780134610993/i9780134610993/1%2C1%2C2%2CB/frameset&FF=i9780134610993&1%2C%2C2



Consultations & Other Academic Support

- Consultations
  - By appointment only
  - Exhaust other channels first
- Canvas discussions
  - Will be answered in reasonable time
- Telegram groups
  - One Telegram group per tutorial class
  - Managed by your tutor

Questions on Administrative Matters?

- Was anything unclear?
- Do you need to clarify anything?
- Ask on Archipelago
  - Specify a question
  - Upvote someone else's question

Invitation Link (Use NUS Email --- starts with E)
https://archipelago.rocks/app/resend-invite/50153102954

What is Artificial Intelligence (AI)?

Artificial Intelligence (in a Nutshell)

- Intelligent mechanisms that solve problems to help humans
  - Programs concerned with human actions / thinking
  - Intelligence assessed based on the mechanism's generality and performance
- Generality
  - More dynamic solutions → able to deal with many cases
    - Example
      - Google DeepMind's AlphaGo, AlphaZero, and MuZero
        https://deepmind.com/research/case-studies/alphago-the-story-so-far (with movies)
- Performance
  - Perform at least as well as humans
    - Not necessarily in the same way as performed by humans (or in nature)
      - e.g., birds versus planes

Kinds of AI

- Strong AI
  - *General* problem-solver
  - Very dynamic programs → solves many problems

Note that solutions are typically assumed to satisfy some performance threshold (e.g., as good as humans)

- Weak / Narrow AI
  - Less dynamic programs → solves fewer problems (typically just 1)
  - Corresponds to most AI work

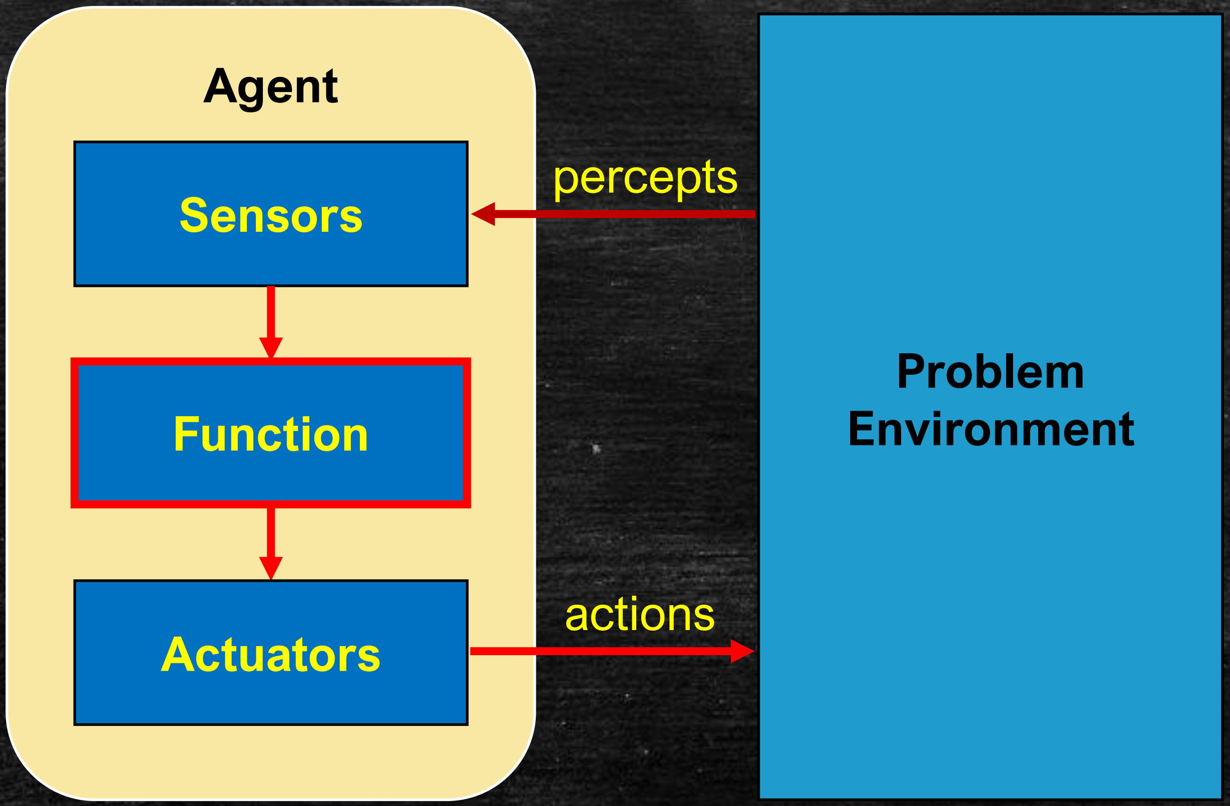
How would you categorise ChatGPT? https://openai.com/blog/chatgpt/

Stronger AI focusing on chat (refer to the Turing Test for an interesting (but older) perspective on general AI)

- Usually focused on Narrow AI
  - Easier to formalise
  - *More on this later …*

Intelligent Agents

Agent Framework



Agent Components

- Sensors and actuators
  - Sensors: what can/should be captured about the environment?
    - Percept data at time step t, p_t
    - Percept sequence, $P = \{ p_1, \ldots, p_t \}$
  - Actuators: how will the agent affect change in the environment?
    - Set of actions, A

- Focus is on the agent function
  - Specify a function f
  - Such that $f : P \rightarrow a_t$
  - Where $a_t \in A$ is the selected action given P

CS3243 focuses on
- Representations for ***P*** and ***A***
- Algorithms that determine f

Rationality & Performance

- Desire a program that works well
  - At least better than humans; ideally optimal
  - Implies a quantifiable objective → performance measure
    - Are the objectives and performance measure aligned?

- Rational agent (function), $f : P \rightarrow a$
  - Given
    - Percept sequence
    - Prior knowledge
    - Set of actions
    - Performance measure

    available data
  - Rational agent optimises performance measure

Note:
do ***not*** assume
agent is omniscient

Why more Narrow AI?
Easier to define the performance measure and thus a rational agent to solve that problem

AI as Search: A First Look

- Goal in AI → determine agent function f
  - $f : P \rightarrow a$
  - $a \in A$

Recall the agent framework
- Agent gets percepts
- Agent function determines action
- Agent enacts action
- Repeat

- Key idea → AI as graph search
  - Each percept corresponds to a state in the problem (state → vertex)
  - Define the desired states → goals
  - After each action, we arrive at a new state (action → edge)
  - Construct a search space (graph)
  - Design and apply a graph search algorithm

(1) Define performance measure and search space

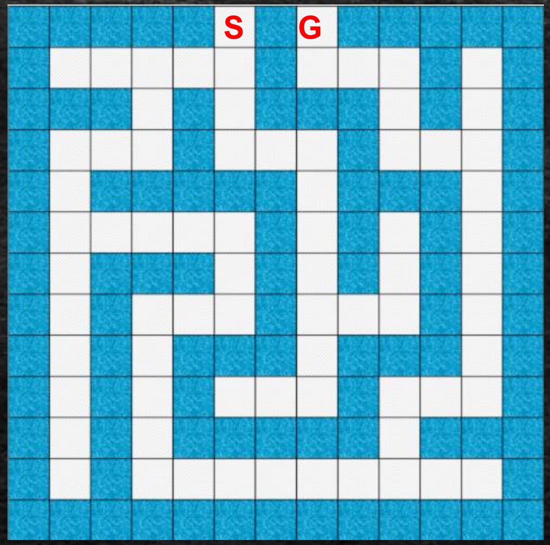
(2) Design search algorithm

First problem we will look at in CS3243 (next week)

– other topics will expand on this idea

An Example Agent

- Problem environment
  - 2-dimensional maze navigation agent
  - P_i : (row, column)
  - A : { ←,↑,→,↓ }



- Agent function f ?
  - Assume map always the same
    - Function: series of if statements
  - Assume map is different each time but remains static during game
    - Function: determined by path planning algorithm (e.g., Dijkstra's)
  - What other possible assumptions?
    - We review this in the next part of the lecture

A Note on Value-Alignment

- Toy problems
  - Objective is clear
  - Performance measure is specific

- Real world problems
  - Objectives not always clear or specific enough → uncertainty in objectives
    - Example: an agent that chooses the best drink for you…
      - Taste versus health
    - Agent may have to learn objectives (e.g., through observation)
  - Not enough time to compute optimal solution → limited rationality

Little to no value-alignment focus in CS324

Questions about the Lecture?

- Was anything unclear?
- Do you need to clarify anything?

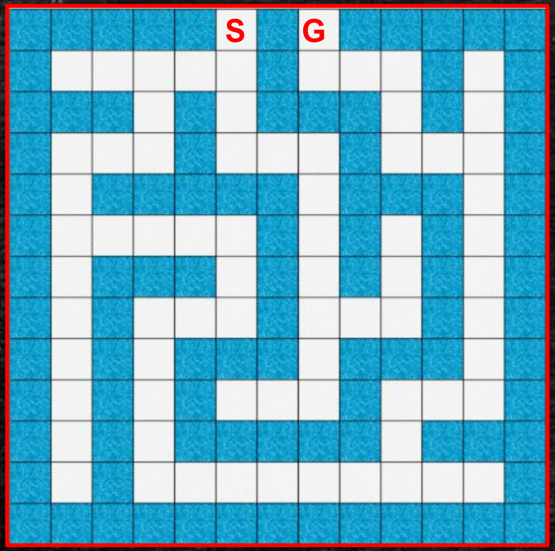
- Ask on Archipelago
  - Specify a question
  - Upvote someone else's question

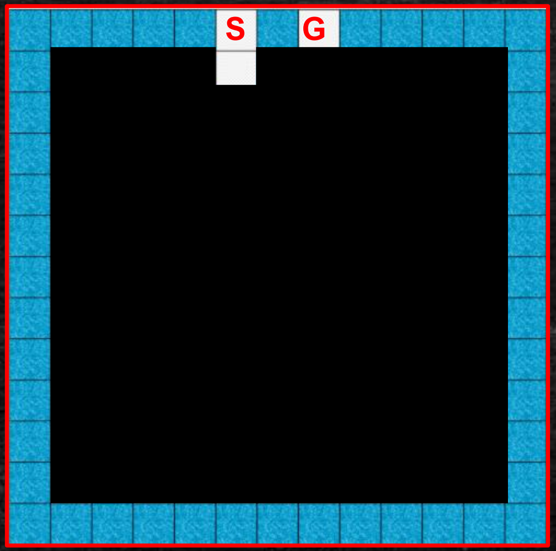
Invitation Link (Use NUS Email --- starts with E)
https://archipelago.rocks/app/resend-invite/50153102954

Problem Environments

Environment Properties

- Fully observable versus partially observable
  - Agent cannot access all information as some cannot be sensed
  - Requires handling uncertainty





Environment Properties

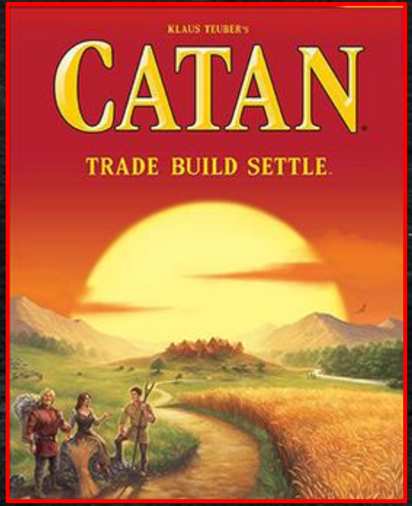
- Deterministic verses stochastic
  - Stochastic → intermediate state cannot be determined based on action taken at a given state
  - Handling uncertainty typically required

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Images taken from Wikipedia

Environment Properties

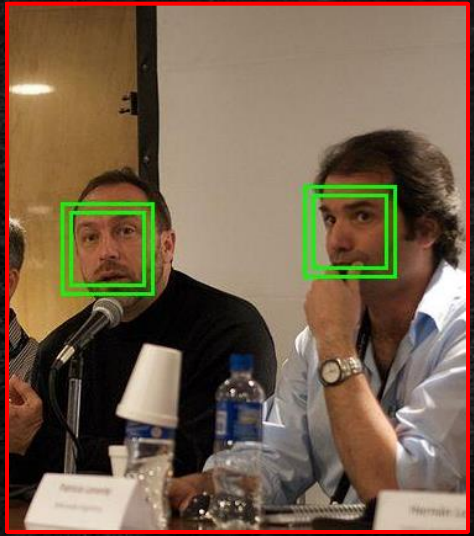
- Stochastic → partially observable?
  - May be fully observable (sense all) but still have randomness with action

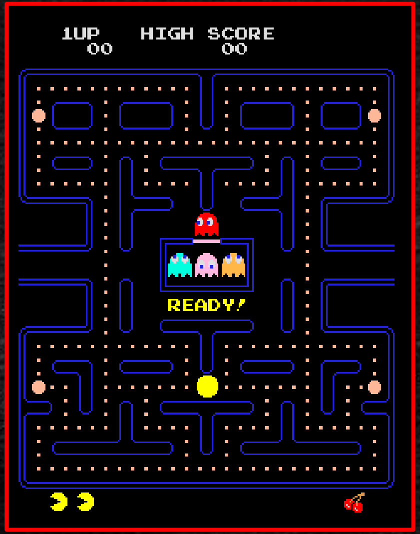


Images taken from Wikipedia

Environment Properties

- Episodic versus sequential
  - Episodic → actions only impact the current state (not those beyond)
  - Sequential → an action may impact all future decisions
  - *Note that it is possible to model an episodic environment into a sequential search space (more on this next week)*





Images taken from Wikipedia

Environment Properties

- Discrete versus continuous
  - Refers to state information, time, percepts, actions

Environment Properties

- Single vs multi-agent
  - Do other entities exist in within the environment that are themselves agents whose actions directly influence the performance of this agent?
    - Chess → opponent is a competitive agent
    - Automated vehicles → other vehicles are cooperating agents

- Known versus unknown
  - Refers to knowledge of the agent/designer (not environment itself)
  - Includes performance measure

- Static versus dynamic
  - Will the environment change while the agent is deciding an action?

Environment Properties

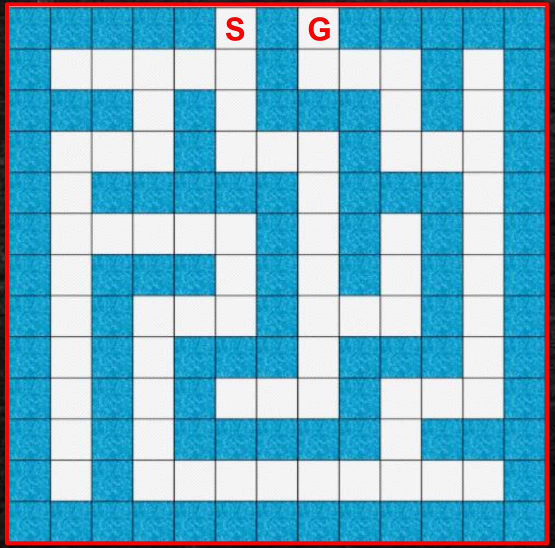
<table>
<tr><th>Property</th><th>CS3243</th><th>Notes</th></tr>
<tr><td>Fully / Partially Observable</td><td>Both</td><td rowspan="2">Latter in Bayesian Networks</td></tr>
<tr><td>Deterministic / Stochastic</td><td>Both</td></tr>
<tr><td>Episodic / Sequential</td><td>Both</td><td></td></tr>
<tr><td>Discrete / Continuous</td><td>Both</td><td>Mostly discrete</td></tr>
<tr><td>Single / Multi-agent</td><td>Both</td><td>Latter in Adversarial Search</td></tr>
<tr><td>Known / Unknown</td><td>Known</td><td></td></tr>
<tr><td>Static / Dynamic</td><td>Static</td><td></td></tr>
</table>

Taxonomy of Agents

Types of Agents

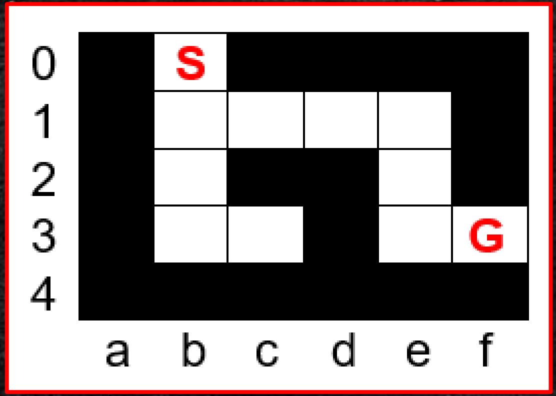
- Reflex agent
  - Uses rules in the form of if-statements to make decisions
  - Direct mapping of percepts to actions
  - Mostly domain specific
  - Impractical with large search spaces

```
if at (0,5): ↓
if at (1,5): ←
if at (1,4): ←
if at (1,3): ↓
if at (1,2): →
if at (1,1): →
...
```



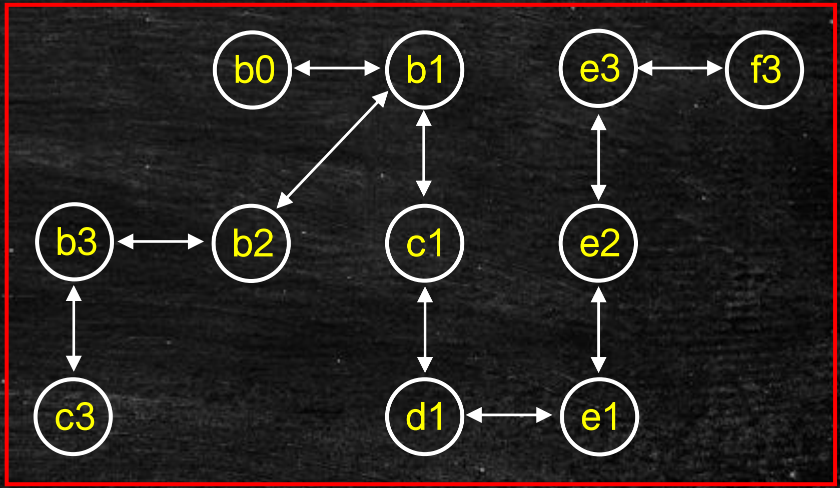
Types of Agents

- Model-based reflex agent
  - Makes decisions based on an internalised model



Adjacency Matrix

State	Actions
b0 (S)	b1
b1	b0, b2, c1
b2	b1, b3
b3	b2, c3
c1	b1, d1
c3	b3
d1	c1, e1
e1	d1, e2
e2	e1, e3
e3	e2, f3
f3 (G)	e3



Types of Agents

- Goal-based and utility-based agents
  - Given
    - State and action representations
    - Definition of goals or utility
  - Determines
    - Sequence of actions necessary to reach goals or maximise utility
    - Or state that satisfies goal conditions or maximises utility

- Learning agents
  - Agents that learn how to optimise performance

Environment Properties

Property	CS3243	Notes
Reflex Agents	Yes	
Model-Based Reflex Agents	Yes	• Logical Agents • Bayesian Networks
Goal-Based and Utility-Based Agents	Yes	• Uninformed / Informed Search • Local Search • Constraint Satisfaction Problems • Adversarial Search
Learning Agents	No	

Questions about the Lecture?

- Was anything unclear?
- Do you need to clarify anything?
- Ask on Archipelago
  - Specify a question
  - Upvote someone else's question

Invitation Link (Use NUS Email --- starts with E)
https://archipelago.rocks/app/resend-invite/50153102954