**COMP30024 - Artificial Intelligence**

**Lecture 1:**

* Lecturer
  + Dr. Sarah Erfani
    - [sarah.erfani@unimelb.edu.au](mailto:sarah.erfani@unimelb.edu.au)
  + Prof. Chris Leckie
    - [caleckie@unimelb.edu.au](mailto:caleckie@unimelb.edu.au)
* Textbook:
  + Artificial Intelligence: A Modern Approach
* Skills
  + Expected data structures and algorithms coding in Python
  + Familiar with found mathematical notation
  + Basic understanding on differential calculus and probability theory
* Assessment
  + 70% final exam
  + 30% assessment
    - Single project submitted in two parts
      * Project is in Python
    - Part 1 due end of week 5
    - Part 2 due end of week 10
  + Need 50% on both to pass hurdle
* Areas using AI
  + Healthcare, Customer Service, Transportation, Manufacturing, Gaming, Smart Homes
* AI are not as smart as you think they are
  + Computer and AI are dumb, they learn to you tell them to do not what they should do
    - Amazon Alexa starts a party calls cops
    - Iphone X hacked by 3D printed
    - Street sign hacks fools self driving cars
  + Leads to a big security risk
* Types of intelligence
  + AI is a science of making computer look like the computers we see in movies
  + What sort of behaviour is identified as intelligent behaviour
* Intelligent behaviour (star wars film)
  + Walking
  + Conversation, natural language
  + Identification of 2nd robot
  + Making good decisions
  + Path finding
  + Social awareness
  + Opinion
  + Inference
  + Emotion
  + Self awareness
  + Long-term planning
  + Computer vision
* Approaches to defining AI
  + Thinking like a human
    - Cognitive modeling by introspection or experimentation
    - Self-awareness is important
    - Humans feel emotions and don't react rationally always
  + Thinking rationally
    - Understand cause and effect
    - Codify rational thinking
    - The study of logic has greatly influenced AI
    - Logic based system does not work well under uncertainties
  + Acting rationally
    - A rational agents performs action which will achieve ones goals
    - Knowledge may not be perfect
* Acting like a human
  + Test for intelligence: Turing Test
    - Aim is for the computer to imitate a human well enough to fool people
    - Thought a computer would pass this test by the end of the century
      * Broken around 2010
    - Suggested major components of AI are:
      * Knowledge, reasoning, language understanding, learning
    - Problem:
      * Turing test is not reproducible, constructive or amenable to mathematical analysis

**Lecture 2:**

* Agent Model
  + Percepts/observations of the environment
    - Made by sensors
  + Action which may affect the environment
    - Made by actuators
  + Environment in which the agent exists
  + Performance measures of the desirability of the environment states
* Agents as functions
  + Agents can be evaluated empirically, sometimes analyzed mathematically
  + Agent is a function from percept sequences to actions
  + Ideal ration al agent would pick actions which are expected to maximize its performance measure
* Agent types
  + Simple reflect agents
    - Assume the world is fully observable
    - Sensory input tells us where we are
  + Model-based agents
    - Have sensory information but need extra knowledge
    - Knowledge inferred from the previous state of the world, current state and what impact our decisions have
  + Goal-based agent
    - Choose the path that takes us closer to our goal
    - What is the world like now
    - What it will be like if I do action
  + Utility-based agents
    - What the world is like now
    - What it will be like when I do decision A
* Environment types
  + Observable
    - Percept contains all relevant information about the world
  + Deterministic
    - Current state of the world uniquely determines the next
  + Episodic
    - Only the current (or recent) percept is relevant
  + Static
    - Environment doesn’t change while the agent is deliberating
  + Discrete
    - Finite number of possible percepts/actions
* Agent model
  + Characterise requirements for an agent in terms of its percept’s, actions, environment and performance measure
* Agent type
  + Choose and justify choice of an agent for a given problem
* Environment types
  + Characterise the environment for a given problem
* Project Assignment
  + Project game:
    - “Watch your back”
    - Dividing line after the 6th row and 2nd row (to rows either side)
    - Each piece has ¾ of the space to move
    - Can’t place a piece in the corners
    - Aim to surround the opponents pieces either horizontally or vertically
    - Friendly corners act as a piece for you
    - 12 pieces (or less are on the board)
    - Can jump over the piece next you
    - Otherwise movement is just up, down, left, right (no diagonal)
    - Can kill multiple pieces with one move
    - Have to make a move if it is possible to move
    - After 64 all the outer layer dies
      * Everything on the edge dies
      * New corners are edge pieces
      * Iterates after 32 moves

**Lecture 3:**

* Problem solving agents
* Abstraction
  + (Abstract) state = set of real states
  + (Abstract) action = complex combination of real actions
  + (Abstract) solution = set of real paths that are solutions in the real world
  + Each abstraction should be ‘easier’ than the original problem
* 8-Puzzle Problem
  + States:
    - Integer locations of tiles
  + Actions
    - Move blank, left, right, up, down
  + Goal test:
    - Goals state (given)
  + Path cost:
    - 1 per move
* State vs. Nodes
  + A state is a (representation of) a physical configuration
  + A node is a data structure constituting part of a search tree
    - Includes parent, children, depth, path cost
  + States do not have parents, children, depth etc.
* Search strategies
  + A strategy is defined by picking the order of node expansion
* Strategies are evaluated along the following dimensions
  + Completeness
    - Does it always find a solution if one exists?
  + Time complexity
    - Number of nodes generated/expanded
  + Space complexity
    - Maximum number of nodes in memory
  + Optimality
    - Does it always find a least-cost solution
* Uninformed search strategies
  + Uninformed strategies use only the information available
    - Breadth-first search
    - Uniform-cost search
    - Depth-first search
    - Depth-limited search
    - Iterative deepening search
* Breadth first search
  + Expand shallowest expanded node
  + Queuing function: put successors at end of queue
  + Time complete, exponential in *d* time, O(*b*d) space, optimal

**Lecture 4:**

* Uniform-cost search
  + Expand least-coast unexpanded node
  + Can get stuck if a low-cost path back to another node
* Depth-first search
  + Expand deepest unexpanded node
  + Add new nodes at the front
  + Depth-first search can perform infinite cyclic excursions
    - Need a non-cyclic search space (repeated state checking)
  + Complete when working in finite space or checking repeated states
    - Otherwise it is not complete
* Depth-limited search
  + Depth-first search with limit depth
* Iterative deepening search
  + Start with a depth of zero and iteratively increase the depth
  + Combination of BFS and DFS
* Bidirectional search
  + Search simultaneously forwards from the start point and backwards from the goal
  + Stop when the two searches meet in the middle
  + Problem
    - Generate predecessors; many goals states; efficient check for node already visited by the other half of the search
* Summary
  + Problem formulation usually requires abstracting away real-world details to define a state space that can be feasibly explored
  + Iterative deepening search uses only linear space and not more time than other uniformed algorithms
* Formal rules of “Watch your back”
  + Coordinate systems (0,0) to (7,7)

**Lecture 5:**

* DFS
  + Add new leaves at the front of the queue
* BFS
  + Add new leaves at the end of the queue
* Best-first Search
  + Estimate the desirability of each node
  + Special cases
    - Greedy Search
    - A\* Search
* Heuristics
  + Used for N-P hard problems
  + Most heuristics are experimental – rule of thumb
    - Solving a jigsaw puzzle by grouping by colour
  + Lose optimality and completeness
  + Need to consider how much time a heuristic can save us
* Greedy search
  + Expands the node that appears to be closest to the goal
  + Not optimal – can be shorter ways
  + Not complete – can get stuck in a loop (unless in a finite space with repeated state checking)
  + Time – but a good heuristic can give dramatic improvement
  + Space – – keeps all nodes in memory
* A\* Search
  + Avoid expanding paths that are already expensive
    - Estimated total cost of path is equal to the cost so far to reach *n* point + the estimate cost to goal from *n*
  + A\* search uses an admissible heuristic
  + If there is no unexpanded node that is less than the current solution then the function will end
  + Complete unless there are infinitely many nodes with
  + Exponential in [relative error in x length of solution)
  + Space – keeps all nodes in memory
  + Optimal – Yes
* Distances
  + Manhattan
    - Square grid that allows 4 directions of movement
  + Diagonal
    - On a square grid that allows 8 directions of movement
  + Euclidean
    - On a square grid that allows any direction of movement

**Lecture 6:**

* Relaxed problems
  + Admissible heuristics can be derived from the exact solution cost of a relaxed version of the problem
* Iterative improvement algorithms
  + In many optimization problems, path is irrelevant
  + What is important is that we get to the goal state
  + In this case we can use iterative improvement algorithms
    - Keep a single ‘current’ state, and try to improve it
* Travelling Salesperson Problem
  + Relaxed problem:
    - Let path be any structure that connects all cities
      * Use minimum spanning tree as heuristic for the TSP