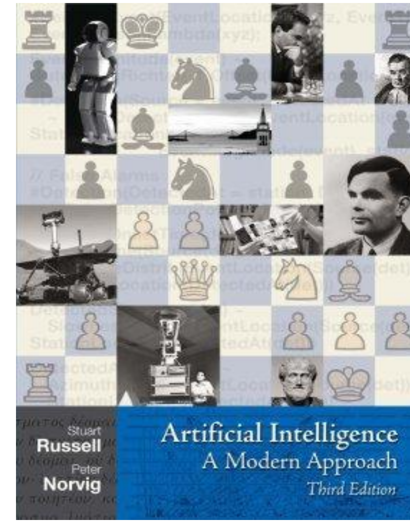
# COSC1125/1127 Artificial Intelligence

## Week 1: Introduction and Search

Readings: [RN2] Sec 3.1–3.5; or [RN3] Sec 3.1–3.4

# Administrivia




- Course Canvas page:
  - https://rmit.instructure.com/courses/49293
- Course Guide:
  - http://www1.rmit.edu.au/courses/COSC11251910
- Course Forum
  - Canvas discussion forums
- Tutorials and Lab sessions (starts from week 2)
- Text Book:

  Russell S. & Norvig P. Artificial Intelligence: A Modern Approach, Prentice Hall, 2009 (Third Edition).
- References / Recourses
- Others

# How to be effective in learning in this course?

- Attend 2hrs lecture per week
  - Get the "big picture"
- Read recommended chapter/sections before class
- 1 hr tutorial, to work on the practical tutorial exercises
  - Group discussion; peer and self assessment.
- 1 hr lab on search animation software (assignment#1), Python (for both assignment#2 and #3), further questions on tutorial exercises, assignments, etc
- Participate actively; bring your questions to the lab.
- Complete practical exercises and move to next set!
- Stay in sync & don't get behind & start assig. early

# Assessments

- Assignment#1: Search (10%; Due Week 3)
- Assignment#2: Connect4 game (15%; Due Week 6)
- Assignment#3: Reinforcement Learning (15%; Due Week 12)

- Class Test Component (Week 7): 10%
- Final Exam: 50%

# Course overview

- Introduction  (week 1)
- Search Strategies (week 2)
- Adversarial Search (week 3)
- Knowledge Representation (week 4 & 5)
- Automated Planning  (week 6)
- Decision Making Under Uncertainty (week 7)
- Reinforcement Learning (week 8)
- Probability (week 9)
- Bayesian Networks (week 10)
- Intelligent Agents (week 11)
- AI at RMIT & Course Review (week 12)

# Why Artificial Intelligence (AI)

It is not just a good science fiction topic.

Successful stories of using AI:

- Expert systems find mineral worth $300,000,000.
- DEC, CXON Configurer Saves $40,000,000 per year.
- Schlumberger, Oil-well log analysis produces $200,000,000 per year.
- Elf Aquitaine, oil drilling diagnosis produces $100,000 per day.
- Digital Equipment, job dispatching saves $26,000,000 per year.
- Sherman/LehmanBros/ American Express: their interest rate swapping services make $1,000,000 per month.
- Coopers/Lybrand's Life Insurance Products save $100,000 per product.
- ANZ Bank's Product Advisor generates $5,000,000 per year.
- Intelligent spidering and retrieval on WWW.

Canadian Pacific: a $250,000 mistake because of not using ES recommendation.

# Where is AI?


WATSON
Google
self-driving car
GARRY KASPAROV
DEEP JUNIOR
INSIDE THE EFFECTS
THE LORD OF THE RINGS
iRobot

# Robots are coming!!

Poster by
AAAI
Association for the Advancement of Artificial Intelligence
AI Magazine
Poster development supported in part by
NSF
Microsoft Research
YAHOO! RESEARCH
Optimizing Paths & Flows
AI & Creative Expression
Planning
Plan Recognition
AI in Art
AI in Music
Autonomous Vehicles & Safety
Robot Guides & Assistants
Augmenting Cognition
Humanoid Robots
Intelligent Tutoring
PV=nRT
Ubiquitous Computing
Social Computing
AI HISTORY TOUR
Gesture Recognition
Multimodal Interfaces
Mixed-Initiative Collaboration
Autonomous Space Exploration
AI and Preferences, Media & Entertainment
Home Robotics
Descartes
Aristotle
Leibniz
Whitehead
Lovelace
Russell
Turing
Robots For Education
Robotic Surgery
Diagnosis
AI LAB
Assistive Technology
Drug Design
Scientific Discovery
MEDICAL CENTER
Search & Retrieval
Security & Privacy
User Modeling
Vehicle Navigation
Handwriting & Sketch Recognition
Recommender Systems & Question Answering
BOOKS & COFFEE
Communications Triage
Machine Translation
Ecocomputing
See the AI timeline and more at
www.aaai.org/AILandscape
The AI Landscape
David Leake, Indiana University, Poster Development Committee Chair
Poster Design: Giacomo Marchesi, www.GiacomoMarchesi.com

# Why AI an Important Course?

- It provides the core knowledge of computer science.
- All serious programmers and software engineers should know about the major artificial intelligence techniques.
- It can be profitable.
- It is also fun.
- It is different to most other subjects.

# What is Intelligence?

- Faculty of understanding
- Capacity to know or apprehend
- Available ability as measured by intelligence tests or social criteria
- Ability to use knowledge in new situations or problems
- Ability to learn
- Ability to plan, to foresee problems
- Ability to use symbols and relationships
- Ability to think abstractly, to work towards a goal
- Ability to perform some of the functions of a computer

# Definitions of AI

- Getting computers to do things which would be considered intelligent if done by people.
- The branch of computer science that is concerned with the automation of intelligent behaviour.
- AI is the study of mental faculties through the use of computational models.
- AI is the computer modeling of human mental abilities.
- AI is the technology of getting computers to do things that seem intelligent.

# Definitions of AI ...

- AI researchers are trying to create a computer that thinks.
- AI is the collection of problems and methodologies studied by AI researchers.
- AI is the study of how to make real computers act like the ones in movies.
- AI is the exploration of the design spaces of intelligences.
- AI is both:
  1. The process of attempting to **understand** intelligence by attempting to model it computationally (scientist' explanation).
  2. The process of attempting to **utilise** intelligence by attempting to model it computationally (engineers' explanation).

# Approaches of AI

There are four categories of approaching AI:

I. Acting humanly

– **The Turing Test** approach: to test whether a system can behave intelligently enough to fool a human interrogator.

II. Thinking humanly

It is the cognitive modeling approach, which is now distinct from AI.

III. Thinking rationally

It is the laws of thought approach.

Direct line through mathematics and philosophy to modern AI.

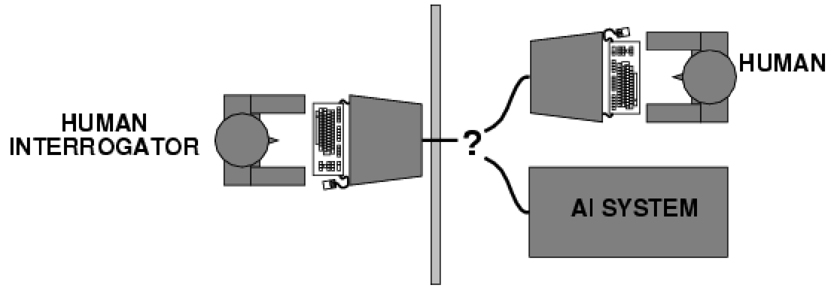
IV. Acting rationally

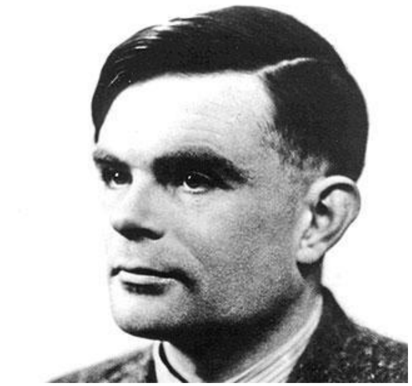
It is the rational agent approach, which is to do the right thing.

The right thing: that which is expected to maximize goal achievement, given the available information.

# Acting humanly: Turing Test

- Turing (1950) "Computing machinery and intelligence":
- "Can machines think?" → "Can machines behave intelligently?"
- Operational test for intelligent behavior: the Imitation Game



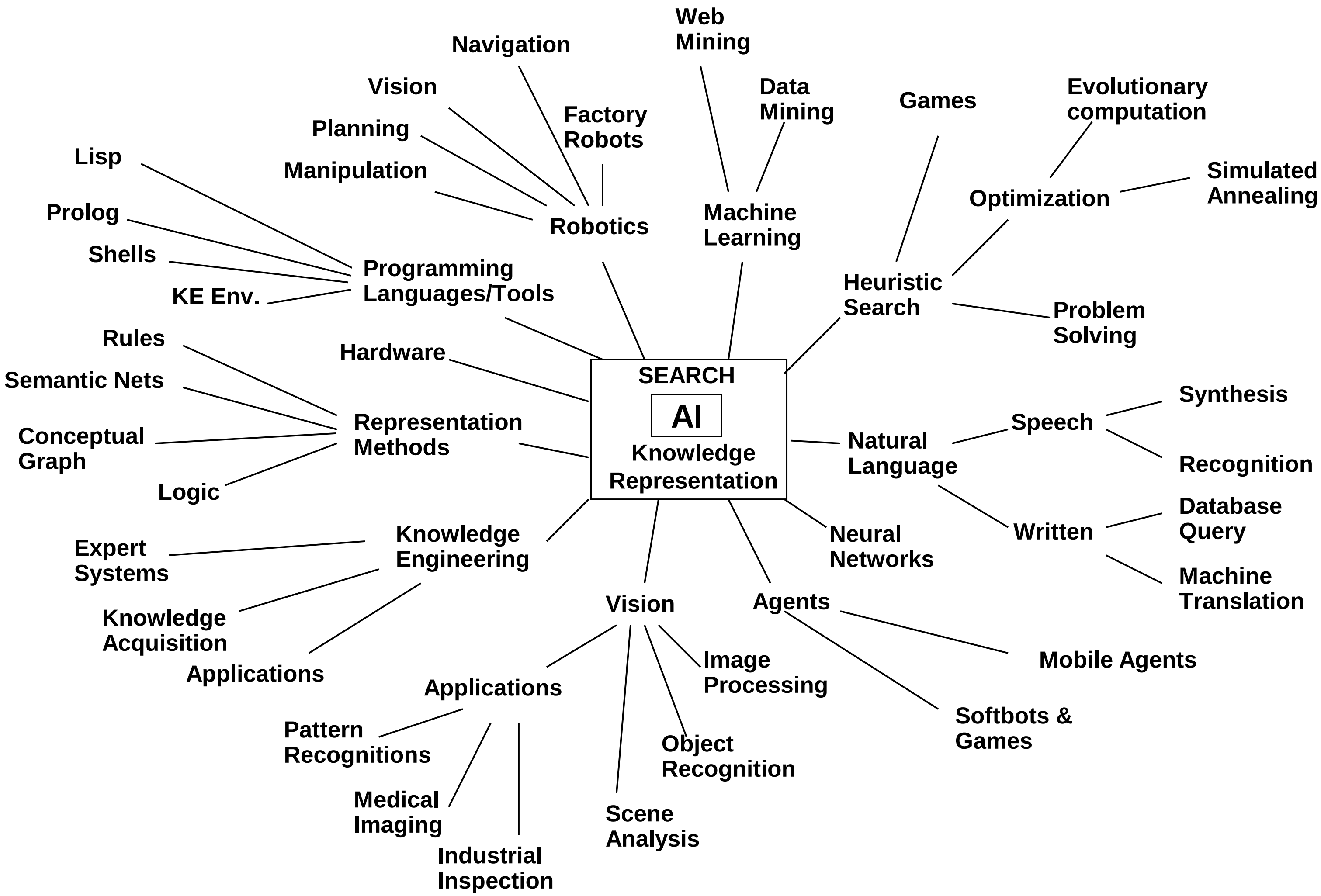



- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes
- Anticipated all major arguments against AI in following 50 years
- Suggested major components of AI: **knowledge, reasoning, language understanding, learning**

# Foundations of AI

| | |
|---|---|
| Philosophy | logic, methods of reasoning |
| | mind as physical system |
| | foundation of learning, language, rationality |
| Mathematics | formal representation and proof |
| | algorithms, computations, decidability, tractability, probability |
| Psychology | adaptation, perception, motor control |
| | experimental techniques |
| Economics | formal theory of rational decisions |
| Linguistics | knowledge representation, grammar |
| Neuroscience | plastic physical substrate for mental activity |
| Control theory | homeostatic systems, stability |
| | simple optimal agent design |

# Map of AI (vic's view)



# History of AI

| | |
|---|---|
| 1943 | McCulloch & Pitts: Boolean circuit model of brain |
| 1950 | Turing's "Computing Machinery and Intelligence" |
| 1952–69 | Look, Ma, no hands |
| 1950s | Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine |
| 1956 | Dartmouth meeting: "Artificial Intelligence" adopted |
| 1965 | Robinson's complete algorithm for logical reasoning |
| 1966–74 | AI discovers computational complexity |
| 1969–79 | Early development of knowledge-based systems |
| 1980–88 | Expert systems industry booms |
| 1988–93 | Expert systems industry bust: "AI winter" |
| 1985–95 | Neural Networks return to popularity |
| 1988– | Resurgence of probability; general increase in technical depth<br>"Nouvelle AI": Alife, Gas, soft computing |
| 1995– | Agents agents everywhere |
| 2003– | Human-level AI back on the agenda |

# State of the Art

- Deep Blue defeated the reigning world chess champion Garry Kasparov in 1997
- Proved a mathematical conjecture (Robbins conjecture) unsolved for decades
- No hands across America (driving autonomously 98% of the time from Pittsburgh to San Diego)
- During the 1991 Gulf War, US forces deployed an AI logistics planning and scheduling program that involved up to 50,000 vehicles, cargo, and people
- NASA's on-board autonomous planning program controlled the scheduling of operations for a spacecraft
- Proverb solves crossword puzzles better than most humans
- Alpha Go winning over human players
- Much more……

# AI at RMIT

| | |
|---|---|
| Lawrence Cavedon | Logic, Intelligent Agents |
| Jeffrey Chan | Machine Learning and Data Mining |
| Vic Ciesielski | Evolutionary Computation, Object Detection |
| | Machine vision, Data Mining, Machine Learning |
| James Harland | Logic and Logic Programming, Intelligent Agents |
| Xiaodong Li | Evolutionary Computation, Machine Learning, |
| | Swarm Intelligence, Multi-objective Optimization |
| Lin Padgham | Description Logics, Intelligent agents, Robo-soccer |
| Sebastian Sardina | Logic and linear logic, Intelligent Agents |
| Andy Song | Evolutionary Computation, Machine Vision, Machine Learning |
| John Thangarajah | Intelligent Agents |
| Tim Wiley | Robotics, Machine Learning |
| Fabio Zambetta | AI in games, Reinforcement Learning |
| Julie Porteous | Interactive Storytelling, AI Planning |

# SEARCH

Chapter 3,4 & 6

# What is Search

**The basic idea of search:**

- before actually doing something in order to solve a problem/puzzle, the possible solutions will be explored.

- Search is an important approach to general-purpose problem-solving.
- Search techniques are the most important and pervasive general AI programming technique.
- Search = "(mental) exploration of possibilities"
- When we make a decision we can not be sure that we have made the best one unless we explore various possibilities (to at least some degree)

# Examples of Search

- Before deciding on a move, a chess player will consider a number of possible moves, weighing up the probable benefits of each.

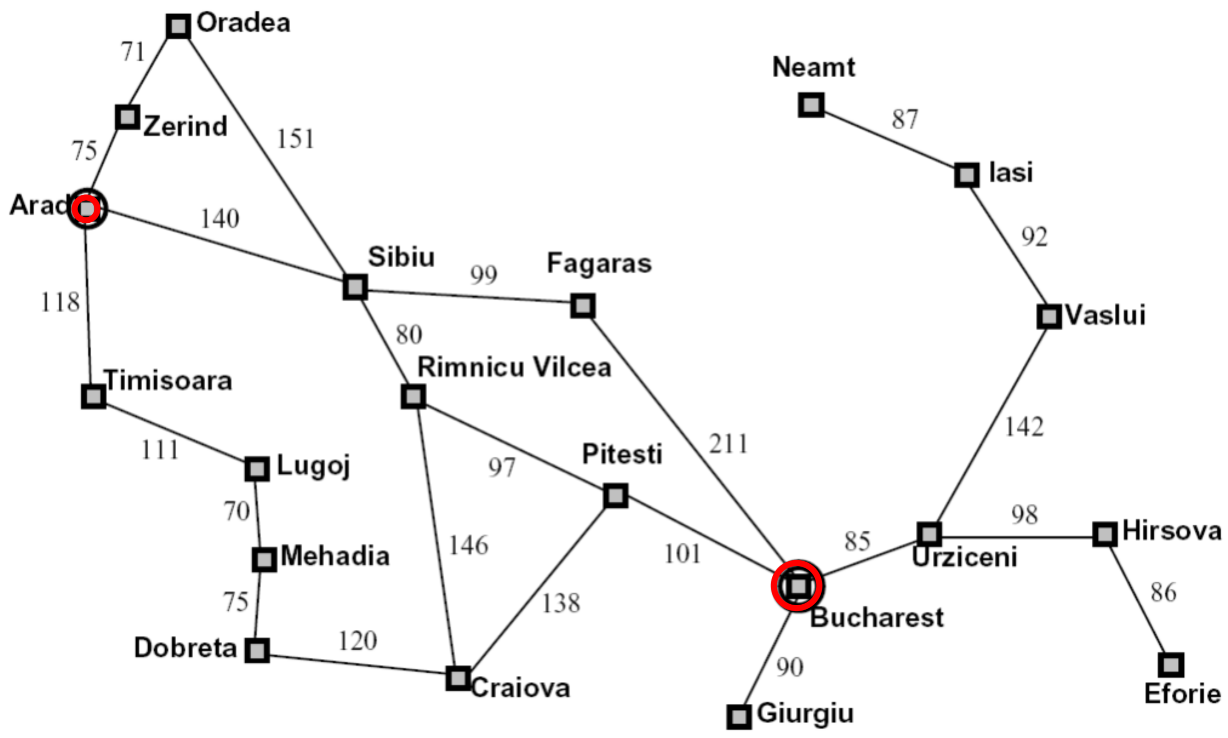
- When trying to solve Rubik's cube, a person may try a few moves "in their head" before deciding on the next move.

- To solve a maze, we may follow some path which leads to a dead-end. We then have to go back to a previous choice-point and try an alternative.

# Examples of Search …

In deciding to travel in Romania, say from Arad to Bucharest, we would look at different routes on a map before choosing one.

(How many possible routes are there?)



# How to Search

- Search is particularly useful for tackling problems for which a “direct” method/algorithm is not known
  - Traveling salesperson problem.
  - Many optimization problems ( e.g best schedule for Olympic Games)

- In the early days of AI, it was thought that searching through possibilities was enough. However, in interesting domains (e.g. chess) there are way too many possibilities.

- We need to search “intelligently”: evaluate possibilities without having to investigate every single possibility.

# Formulating Search Problems

All search problems can be cast into the following general form:

- **Initial State**, such as
  - *Starting city for a route* [Traveling Problem]
  - *A jumbled Rubik's cube* [Rubik's Cube Problem]

- **Goal State** (OR a test for goal state), such as
  - *Destination city* [Traveling Problem]
  - *The solved Rubik's cube* [Rubik's Cube Problem]

- **The permissible operators**, such as
  - *Go to city X* [Traveling Problem]
  - *Rotate subcube ……* [Rubik's Cube Problem]

# State and State Space

Initial state and Goal state are essential in search. So what is a state?

A **state** is one status of the problem. It captures all **relevant** information about the problem at that status, such as

- *A partial route (from starting city to city $X_1$, to city $X_2$…to city X)*

  [Traveling Problem]

- *A unsolved Rubik's cube (the orientation of all sub-cubes of the cube)*

  [Rubik's Cube Problem]

**State space** is the collection of all states – all possible statuses of the problem. Sometimes it is also known as search space.

- *All possible routes* [Traveling Problem]
- *All possible combinations of sub-cubes* [Rubik's Cube Problem]

# Types of Search Problems

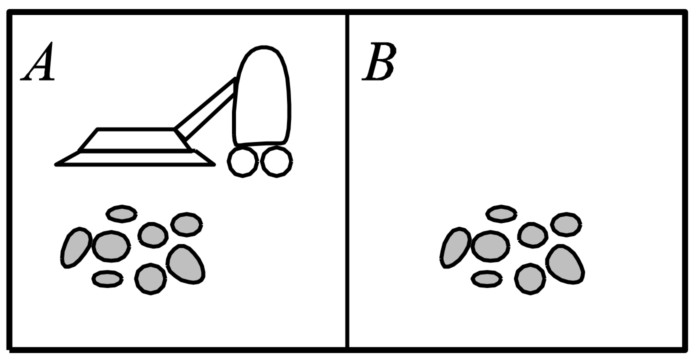
Search problems can be categorized into several groups:

- **Deterministic, fully observable environment** ⇒ *single-state problem*
  - Agent knows exactly which state it will be in.
  - Solution is a sequence of actions.
- **Non-observable environment** ⇒ *conformant problem*
  - Agent know it may be in any of a number of states.
  - Solution, if any, is a sequence of actions.
- **Nondeterministic and/or partially observable environment** ⇒ *contingency problem*
  - Percepts provide new information about current state.
  - Solution is a tree or policy.
  - Often interleave search and execution.
- **Unknown state space** ⇒ *exploration problem ("online")*

# Example: vacuum world

It is a made-up world:

- There are only two squares,
- and an "intelligent" vacuum cleaner which can perceive which square it is in and whether there is dirt in the square.
- The cleaner can choose to move left, move right, suck up the dirt, or do nothing.
- The goal is to have no dirt in both squares.



# Example: vacuum world…

There are eight possible states in the state space.

The goal state is state 7 or 8 (a clean world).

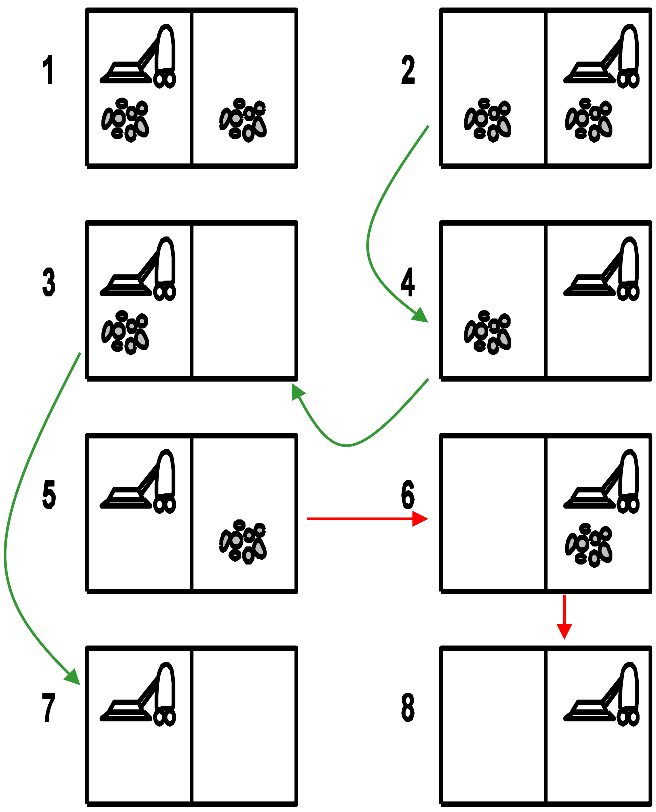
The operators are “Move Left”, “Move Right” and “Suck”.

If the initial state is state 5, how to solve this problem?

(**The solution path**, the path from the initial state to a goal state, is marked in red.)

What if the initial state is state 2 ?

(The solution path is marked in green.)



# Example: Cannibals and Missionaries (C & M)

*Three cannibals and three missionaries come to a crocodile infested river. There is a boat on their side that can be used by one or two persons. If cannibals outnumber the missionaries at any time, the cannibals eat missionaries. How can they use the boat to cross the river so that all missionaries survive?*

The initial state is: (3, 3, LEFT , 0, 0)

The goal state is: (0, 0, RIGHT, 3, 3)

Number of **Cannibals** on **left** side of the river.

Number of **Missionaries** on **left** side of the river.

Shows the boat is on which side of the river.

Number of **Cannibals** on **right** side of the river.

Number of **Missionaries** on **right** side of the river.

https://en.wikipedia.org/wiki/Missionaries_and_cannibals_problem

# Operators for C & M Problem

A legal move is one which moving up to two people to the opposite bank (such that cannibals don't outnumber missionaries on both sides).

Some possible moves are:

(3, 3, LEFT , 0, 0) -> ( 2, 2, RIGHT, 1, 1)

(2, 2, RIGHT, 1, 1) -> ( 2, 3, LEFT , 1, 0)

An operator can be written in the format of **Move-1m1c-lr** (Move **1 m**issionary and **1** **c**annibal from **l**eft bank to **r**ight bank)

**Move-1m1c-lr** brings state (*cleft, mleft, boatpos, cright, mright*) to

state (*cleft–1, mleft –1, RIGHT,cright +1, mright +1* ).

Note: operator **Move-1m1c-lr** has preconditions which are *boatpos* = LEFT, *cleft* >= 1, *mleft* >= 1 and cannibals don't outnumber missionaries on both sides after the move.

[ Question: how to represent the last precondition? ]

So can this operator be applied to the state (2, 2, RIGHT,1, 1) ?

# Operators for C & M Problem…

Another operator **Move-2m-rl** (Move **2 m**issionaries from **r**ight bank to **l**eft bank)

**Move-2m-rl** brings state (*cleft, mleft, boatpos, cright, mright*) to

state (*cleft, mleft+2, LEFT, cright, mright −2*).

For example, from state (1, 1, RIGHT , 2, 2) to a new state ( 1, 3, LEFT, 2, 0)

The preconditions of this operator are:

- *boatpos* = RIGHT,
- *mright* >= 2,
- *cleft* <= *mleft* +2
- (*cright* < = *mright* −2 ) OR *mright* = 2

So this operator could not be applied to the state (2, 2, RIGHT,1, 1).

# Operator Set of C & M

A complete set of operators would be:

**Move-1m1c-lr**

**Move-1m1c-rl**

**Move-2c-rl**

**Move-2c-lr**

**Move-2m-rl**

**Move-2m-lr**

**Move-1c-rl**

**Move-1c-lr**

**Move-1m-rl**

**Move-1m-lr**

Question: give the preconditions and the output state for each of these operator assuming the predecessor state is (*cleft, mleft, boatpos, cright, cright*).

- State Space of C & M

Part of the state space of C & M problem can be represented as the following graph.

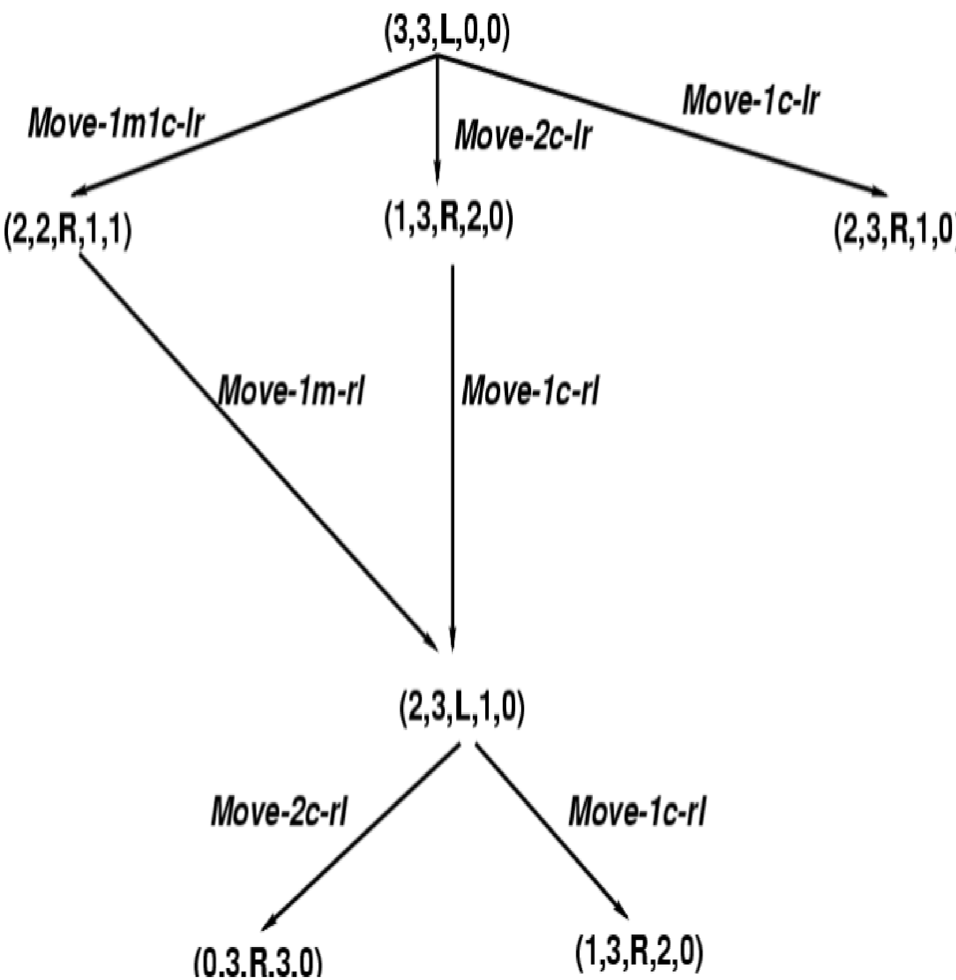
# State Space of C & M

Putting in links to nodes already visited clutters up a state space graph.

To leave out these links, a list can be kept to record the states already visited.

The state space graph of cannibals and missionaries problem can be redrawn like the figure at right.

Question: can you draw the complete state space graph, and what is the solution path?




A Youtube shows how to find a solution: [https://www.youtube.com/watch?v=W9NEWxabGmg](https://www.youtube.com/watch?v=W9NEWxabGmg)

# State Space Search

The previous slide shows an example of state space search, search a solution in the state space.

- State space search involves the use of a graph to keep track of the relationships between states.
- Each node of the graph represents a state of the problem.
- Each arc in the graph represents the application of an operator in the search process.

The solution is

- sometimes the sequences of operators which transform the start state to the goal state.
- sometimes the actual goal state.

# State Space Representation

When defining a problem as state space search, the representation of states is critical.

The representation of a state must encapsulate **all the relevant** information necessary to decide "what to do next", and **none of the irrelevant** information.

- In the C & M problem position of boat is relevant.
- Crocodiles or which way the river is flowing are irrelevant.

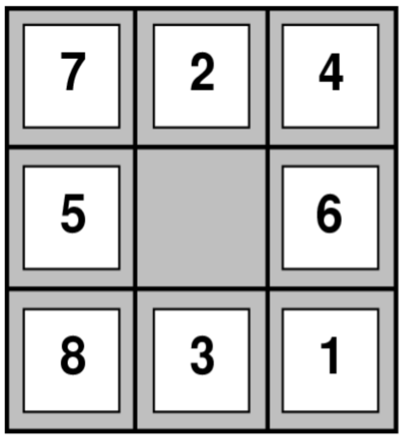
The initial state, the goal state, the operators should be defined in the representation.

Representation varies from problem to problem. A clever representation may reduce computation.

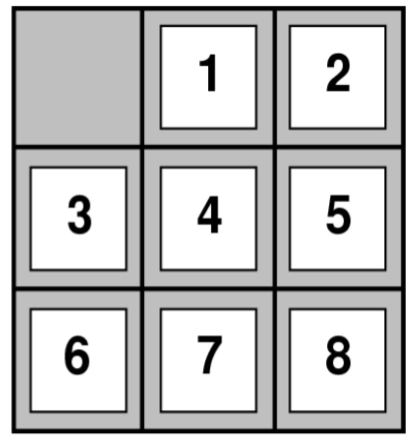
# Example: the 8-puzzle

This puzzle consists of a 3x3 grid with 8 consecutively numbered tiles arranged on it. Any tile adjacent to the blank grid can be moved to it.

One can choose the initial state and the goal state:




Start State




Goal State

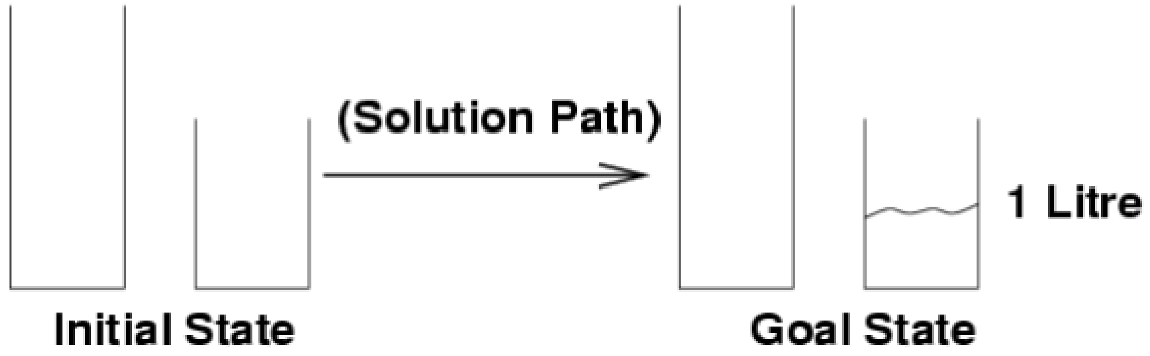
The states could be represented as integer locations of tiles.

The operators could be represented as swap blank with left tile, right tile, upper tile or lower tile.

Question: can the full state space be drawn?

# Example: water jugs problem

We have one 3 litre jug, one 5 litre jug and an unlimited supply of water. The goal is to get <u>exactly</u> one litre of water into either jug. Either jug can be emptied or filled, or poured into the other.




Question: how would you represent

- the initial state
- the goal state
- the operators
- Can the full state space be drawn ?

[* It is one of the tutorial questions. ]

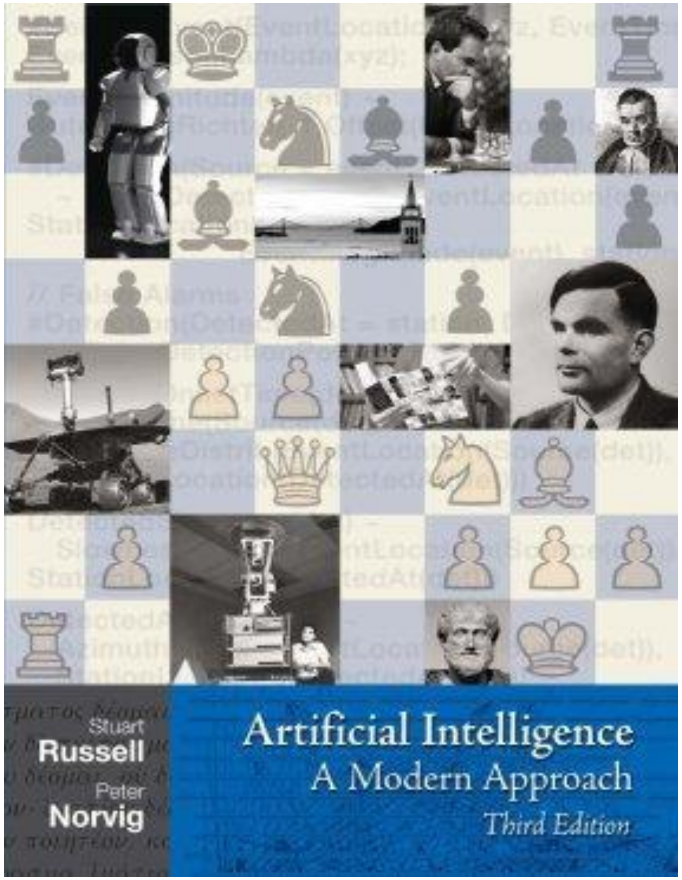
# Performing State Space Search

State space search involves finding a path from the initial state of a search problem to a goal state in the state space.

- To do this, we build a *search graph*, starting from the initial state (or the goal state)
- We *expand* a state by apply the operators to that state, generating **ALL** of its successor states.
- The successors are placed in the next level down of the search graph.
- Such a search graph is actually a tree. It is also known as ***Search Tree***.
- The order in which we choose states for expansion is determined by the ***search strategy***.
- Different strategies result in (sometimes massively) different behaviour.

# Summary

- No tutorials this week
- No labs this week
- Start thinking about how to go about Assignment#1
- Start reading text book
- Reading for this week is chapter 1 of the text book




**Acknowledgement**: the slides were developed based on notes from Russell & Norvig, by several computer science staff members over the years.