Unit-1- Introduction to AI

AI techniques, Problem solving with AI, AI Models, Data acquisition and learning aspects in AI, Problem solving- Problem solving process, formulating problems, Problem types and characteristics, Problem space

and search, Toy Problems – Tic-tac-toe problems, Missionaries and Cannibals Problem, Real World Problem – Travelling Salesman Problem

Problem-solving

Introduction to AI

- According to Rich and Knight,

  "AI is the study of how to make computers do things which at the moment, people do better"

- Branch of science which makes machine intelligent enough in comparable to people

- Involves the study of Psychology (human), giving human reactions to a machine and also mathematical modelling to optimize

Introduction to AI

- Horizon of AI includes,
  - Knowledge Transmission
  - Knowledge Representation
  - Automated Reasoning

- Machines should act rationally

What is intelligence?

➢ Intelligence is an umbrella term used to describe a property of the mind that encompasses many related abilities, such as the capacities

–to reason,
–to plan,
–**to solve problems,**
–to think abstractly,
–to comprehend ideas,
–to use language, and
–to learn.

Intelligence can be defined as the ability for solving problems

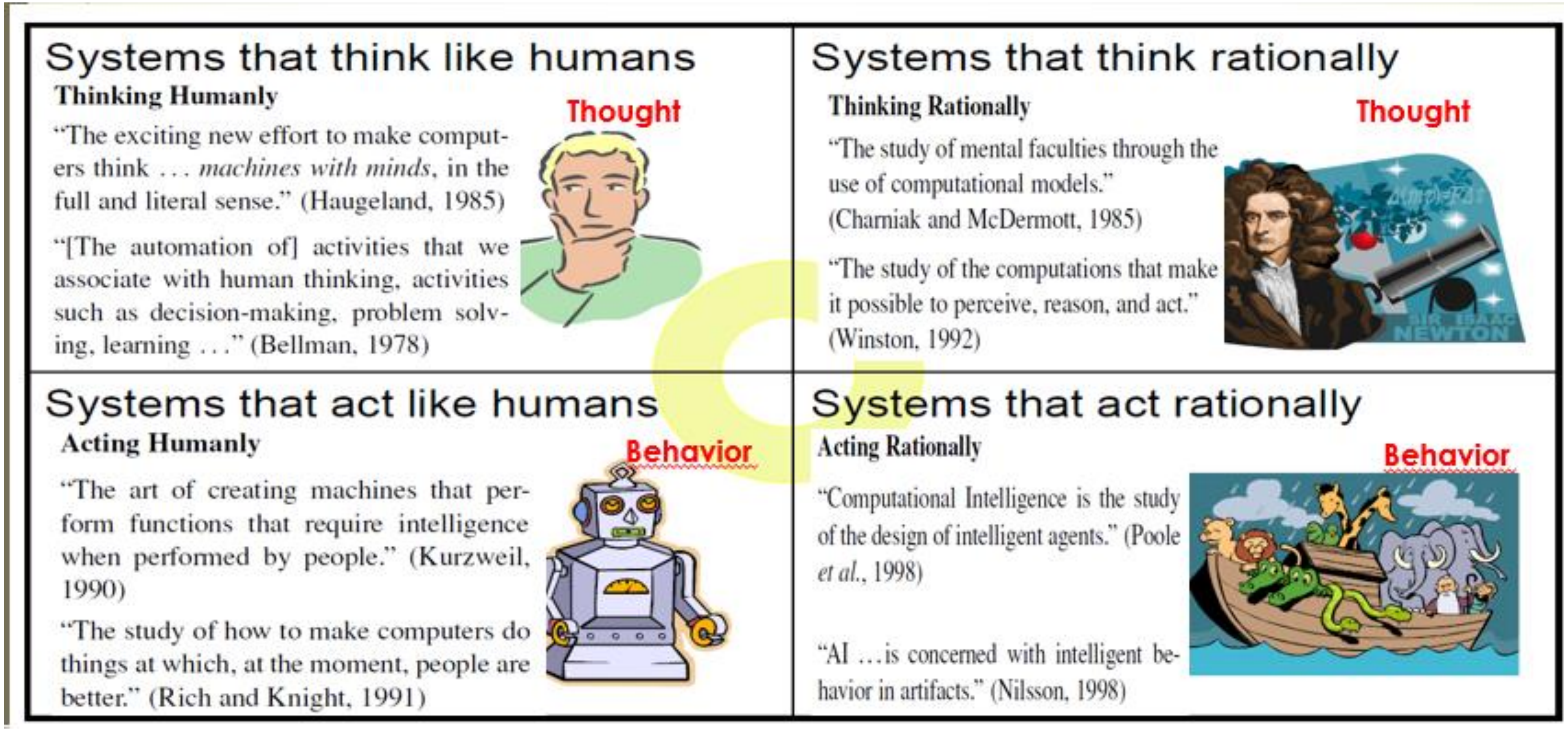
- Problem solving is to find the “best” solution in the problem space.
  - Reasoning is to interpret or justify solutions or sub-solutions.
  - Planning is to find ways for solving the problem.
  - Thinking abstractly is to simulate the problem solving process inside the system (brain).
  - Idea/language comprehension is a way (or means) for data/problem/knowledge representation;
  - Learning is the process to find better ways for solving a problem (or a class of problems).

What is AI ?

- Textbooks often define artificial intelligence as “the study and design of computing systems that perceives its environment and takes actions *like human beings*”.
- The term was introduced by John McCarthy in 1956 in the well-known Dartmouth Conference.
- AI is defined as a system that possesses at least one (not necessarily all) of the abilities mentioned :
    - Reasoning
    - Planning
    - Thinking
    - Idea/language comprehension
    - Learning

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY

A rough classification of AI(from “Artificial Intelligence: A Modern Approach”)



Systems that think like humans **Thinking Humanly** Thought “The exciting new effort to make computers think . . . *machines with minds*, in the full and literal sense.” (Haugeland, 1985) “[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . .” (Bellman, 1978)	**Systems that think rationally** **Thinking Rationally** Thought “The study of mental faculties through the use of computational models.” (Charniak and McDermott, 1985) “The study of the computations that make it possible to perceive, reason, and act.” (Winston, 1992)
Systems that act like humans **Acting Humanly** Behavior “The art of creating machines that perform functions that require intelligence when performed by people.” (Kurzweil, 1990) “The study of how to make computers do things at which, at the moment, people are better.” (Rich and Knight, 1991)	**Systems that act rationally** **Acting Rationally** Behavior “Computational Intelligence is the study of the design of intelligent agents.” (Poole *et al.*, 1998) “AI . . . is concerned with intelligent behavior in artifacts.” (Nilsson, 1998)

Stuart J. Russell, Peter Norwig , Artificial Intelligence –A Modern approach

Thinking humanly: cognitive modeling

Determining how humans think

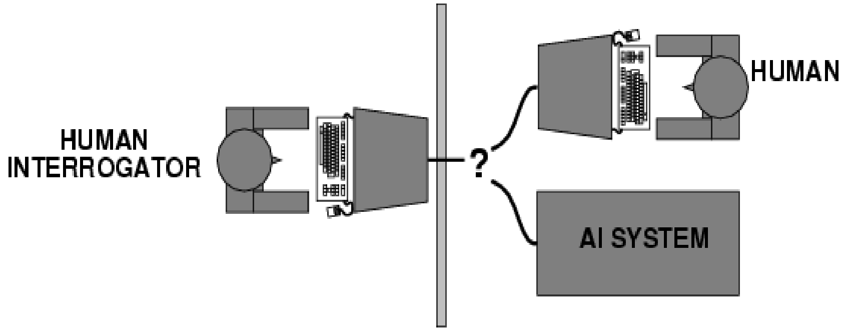
- through introspection—trying to catch our own thoughts as they go by
- through psychological experiments

Express the theory as a computer program

- program's input/output and timing behavior matches human behavior

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



The computer would need to possess the following capabilities:

- **natural language processing** to enable it to communicate successfully in English (or some other human language);
- **knowledge representation** to store information provided before or during the interrogation;
- **automated reasoning** to use the stored information to answer questions and to draw new conclusions;
- **machine learning** to adapt to new circumstances and to detect and extrapolate patterns.

To pass the total Turing Test, the computer will need

- **computer vision** to perceive objects
- **robotics** to move them about.

Thinking rationally: "laws of thought"

- Aristotle: what are correct arguments/thought processes?
- Several Greek schools developed various forms of *logic*: *notation* and *rules of derivation* for thoughts; may or may not have proceeded to the idea of mechanization
- Direct line through mathematics and philosophy to modern AI
- Problems:

1. Not all intelligent behavior is mediated by logical deliberation
2. What is the purpose of thinking? What thoughts should I have?

Acting rationally: rational agent

- Rational behavior: doing the right thing
- The right thing: that which is expected to maximize goal achievement, given the available information
- An **agent** is just something that perceives and acts
- Doesn't necessarily involve thinking – but thinking should be in the service of rational action

Rational agents

- An agent is an entity that perceives and acts
- Abstractly, an agent is a function from percept histories to actions:

$$[f: P^* \rightarrow A]$$

- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance
- computational limitations make perfect rationality unachievable

  → design best program for given machine resources

AI – History and Foundations

- AI entered mainstream before 60 years
- Defined as Statistics, Analysis of patterns and use of formal systems
- In 1940 **Zuse** developed artificial chess playing using high level language called Plankalkul.
- **Leibniz** developed a language for reasoning using symbols

AI – History and Foundations

- **Alan Turing**, British mathematician and WWII code-breaker, is widely credited as being one of the first people to come up with the idea of machines that think in 1950.
- He even created the Turing test, which is still used today, as a benchmark to determine a machine's ability to "think" like a human.
- Though his ideas were ridiculed at the time, they set the wheels in motion, and the term **"artificial intelligence"** entered popular awareness in the mid- 1950s, after Turing died.

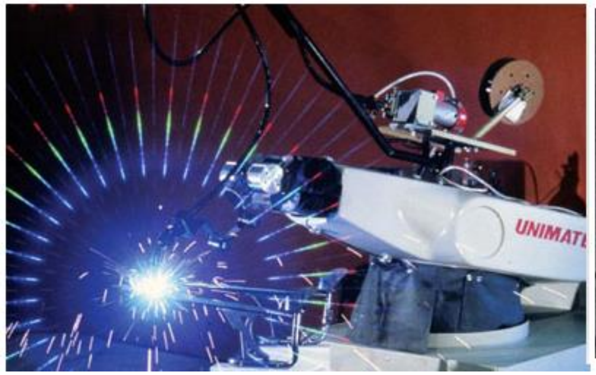
AI – History and Foundations

- **Isaac Asimov**, was an American writer and professor of biochemistry at Boston University.
- The Three Laws of Robotics. The rules were introduced in his 1950 short story "Runaround" and "I,Robot"

- First Law
  - A robot may not injure a human being or, through inaction, allow a human being to come to harm.
- Second Law
  - A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
- Third Law
  - A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

AI – History and Foundations

- 1951 – First AI based program was written
  - a checkers-playing program written by Christopher Strachey and a chess-playing program written by Dietrich Prinz.
- 1955 – First self learning game playing
  - competing against human players in the game of Checkers
- 1959 – MIT – AI based lab setup
- 1961 – First Robot is introduced into GM's assembly line



AI – History and Foundations

- 1964 – First demo of AI program which understand natural language
- 1965 – First chat bot Eliza was invented
- 1974 – First Autonomous vehicle is created
- 1989 – Carnegie Mellon created the first autonomous vehicle using neural networks
  - ALVINN, which stands for Autonomous Land Vehicle In a Neural Network

AI – History and Foundations

- 1996 – IBM’s deep blue – chess playing game
    - Deep Blue won its first game against world champion Garry Kasparov in game one of a six-game match on 10 February 1996.
- 1999 – Sony introduces AIBO – self learning entertaining robot

AI – History and Foundations

- 1999 – MIT AI lab – first emotional AI is demonstrated
- 2004 – DARPA – introduce first vehicle challenge
- 2009 – Google – stared to build a self driving car
- 2010 – Narrative Science AI demonstrate ability to write reports
- 2011 – IBM watson beats jeopardy champions – quiz game
    - the Watson computer system competed on Jeopardy against champions Brad Rutter and Ken Jennings, winning the first place prize of $1 million.

AI – History and Foundations

- 2011 – Google now and Cortana becomes mainstream
- 2016 – Stanford issues the AI 100 reports
- 2016 – UC Berkley launces the centre for human compatible AI

AI – History and Foundations

- AI was started long back

- This is made as possible currently
  - Big data
  - Computing Power

Birth of AI

- Initially, AI dealt with simple reasoning and reaction problems. It requires only very less knowledge base

- Over the period of time, domains like speech recognition, image processing and medical image diagnosis had been added under AI

- Later, AI started to handle complex task with more knowledge base

Examples

- Building intelligent systems,
  - Water tap

    When tank gets filled, switch off
  - Washing machines

    Stops water after reaching particular level

    Fuzzy logic takes necessary amount of water only
  - Traffic control

    Automatically, dynamically adjust signal timing, info to nearby signals, etc…

Examples

- So, the basics of AI,
  - **Machine has to learn**
  - **Machine learn from given knowledge**
  - **Knowledge should be properly represented**

Advantages of Artificial Intelligence

- more powerful and more useful computers
- new and improved interfaces
- solving new problems
- better handling of information
- relieves information overload
- conversion of information into knowledge

The Disadvantages

- increased costs
- difficulty with software development - slow and expensive
- few experienced programmers
- few practical products have reached the market as yet.

AI Techniques

- Types of problem solved,
  - Various day to day problems
  - Identification and Authentication in security
  - Various classification problems in decision making
  - Interdependent and cross domain problem

Data Acquisition and Learning Aspects in AI

- Knowledge discovery
  - Data Mining
  - Machine learning
- Computational learning theory
  - Study and analyse of algorithms
  - Lot of mathematical models
- Neural and Evolutionary computation
  - Neural: mimics the neural behaviour of human beings
  - Evolutionary: Biological behaviours
- Intelligent agents and multi-agent systems
  - Agent: a software which is flexible and supports users

Problem solving

- Given situation -> Desired situation

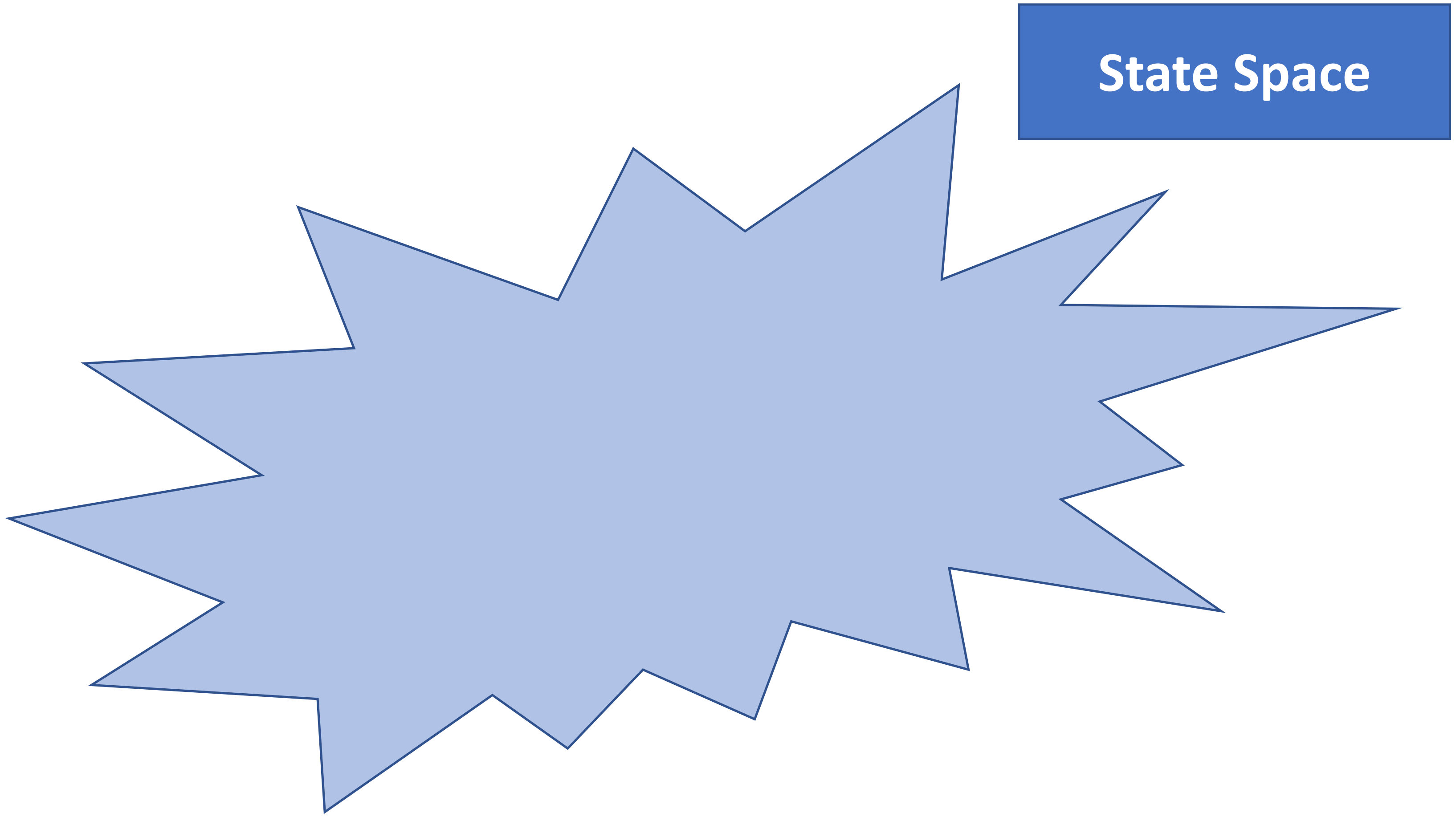
- Task of AI – to perform series of action to move from given situation to desired situation

Types of problem solving

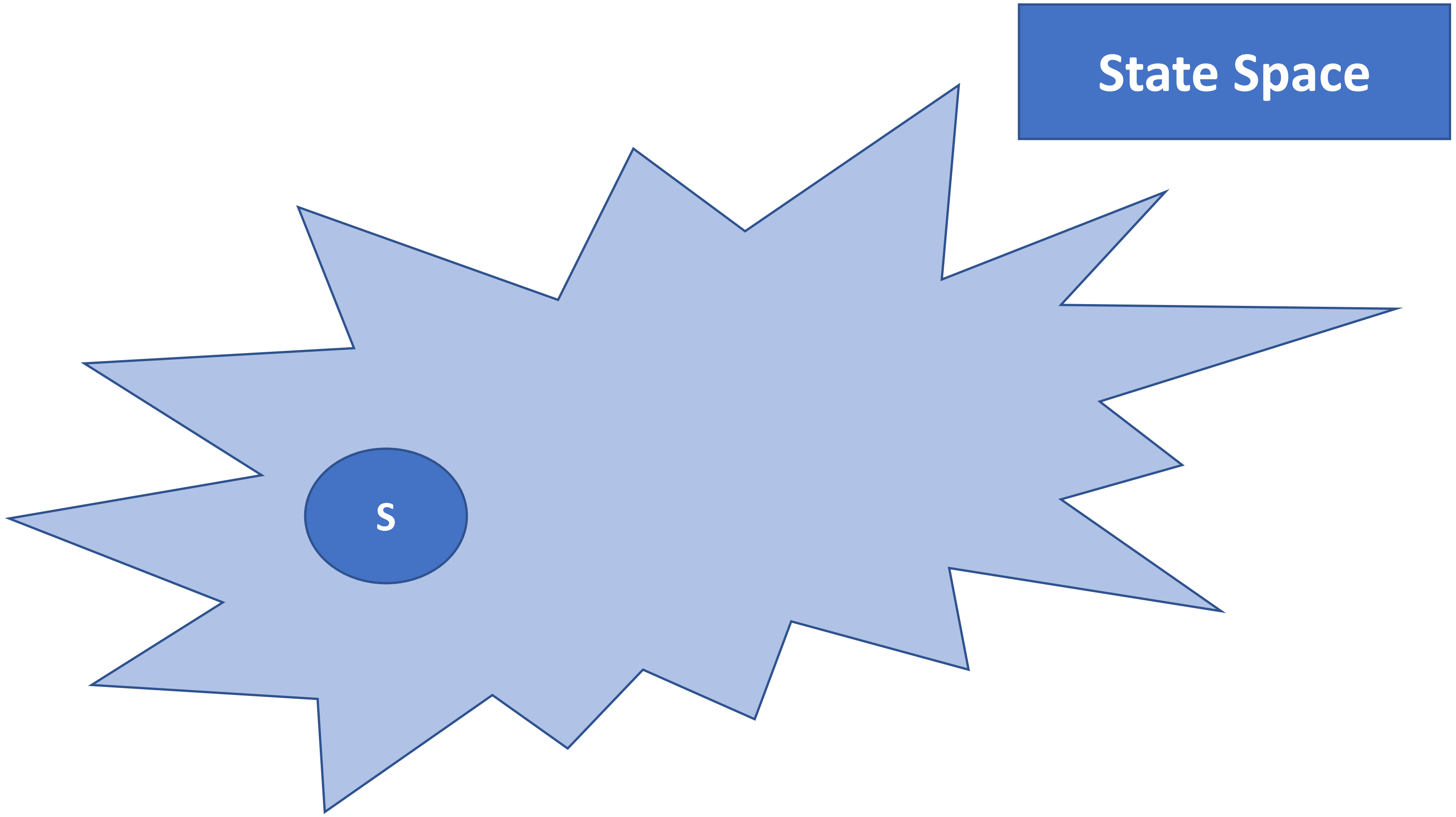
- Knowledge based
  - Memory based
  - Rule based

- Search based

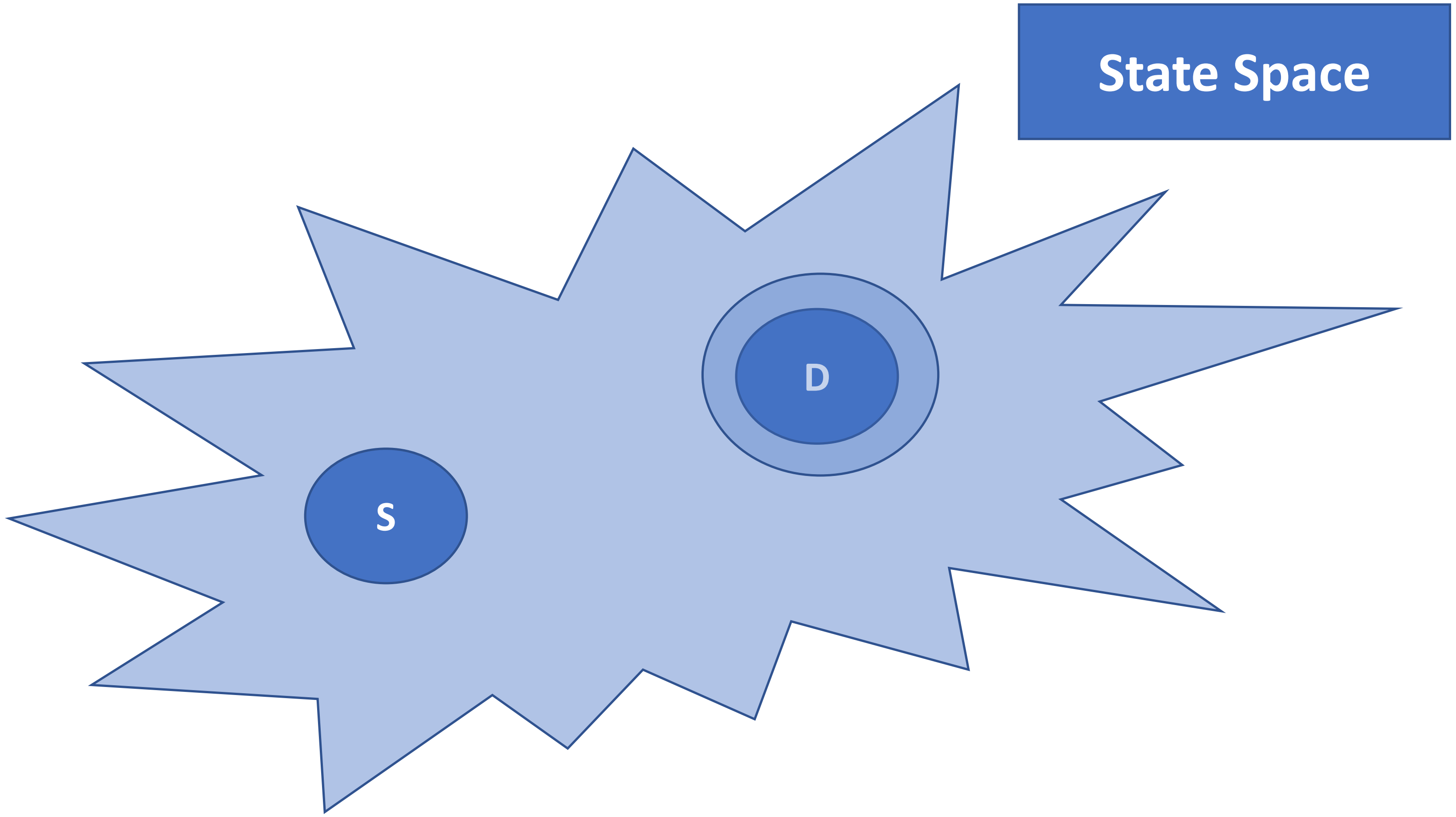
Search based method – state space



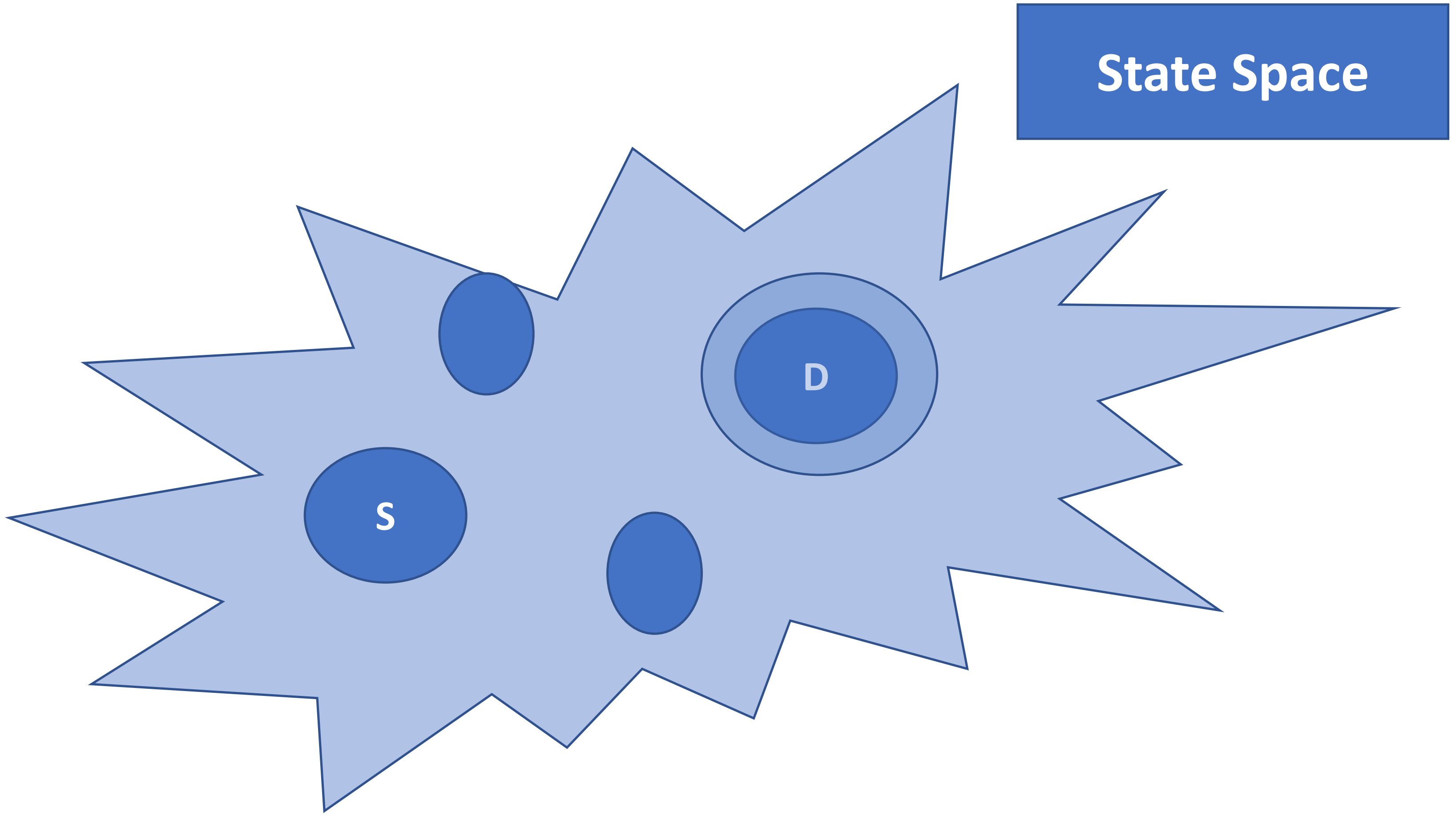
Search based method – state space



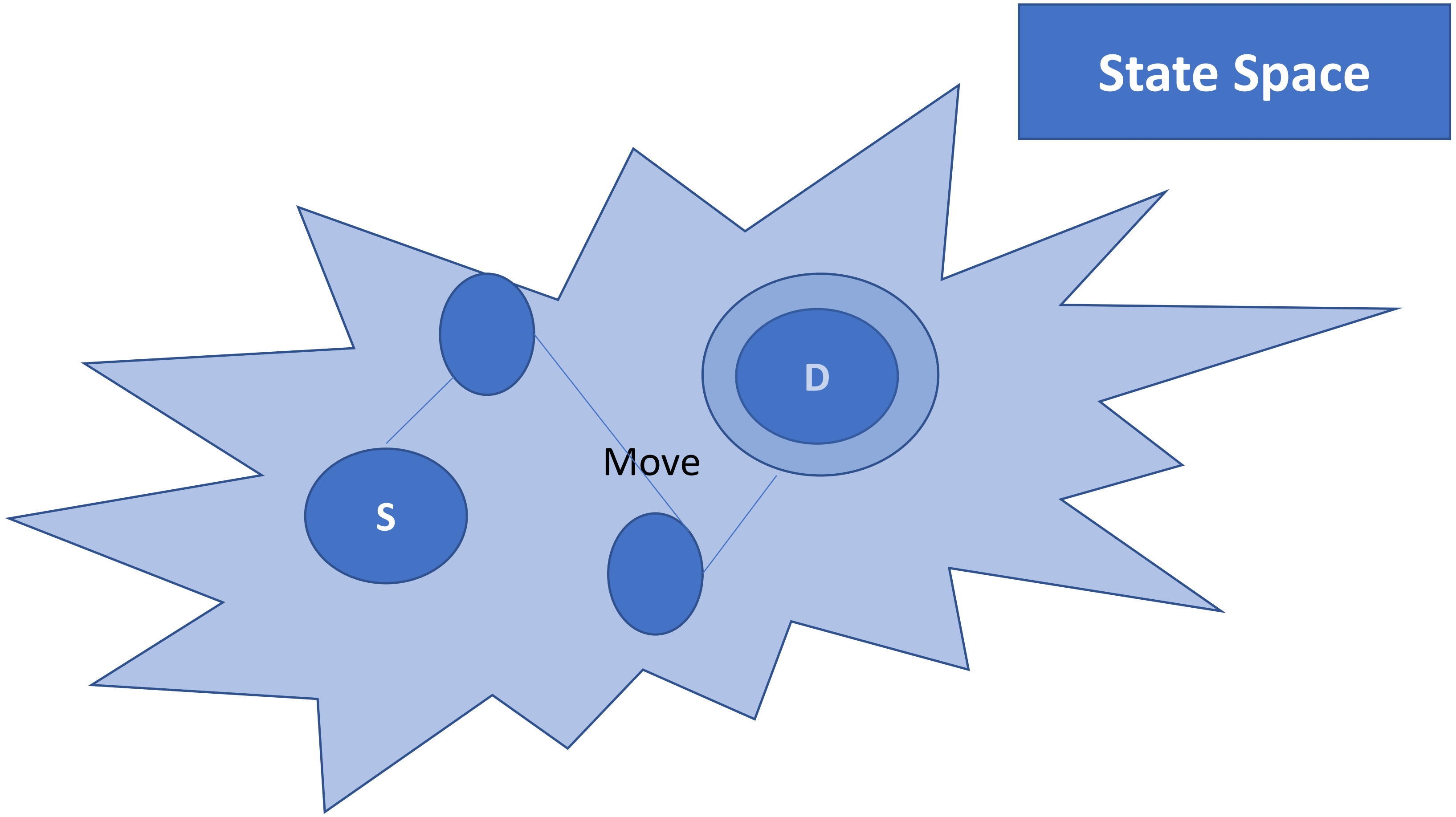
Search based method – state space



Search based method – state space

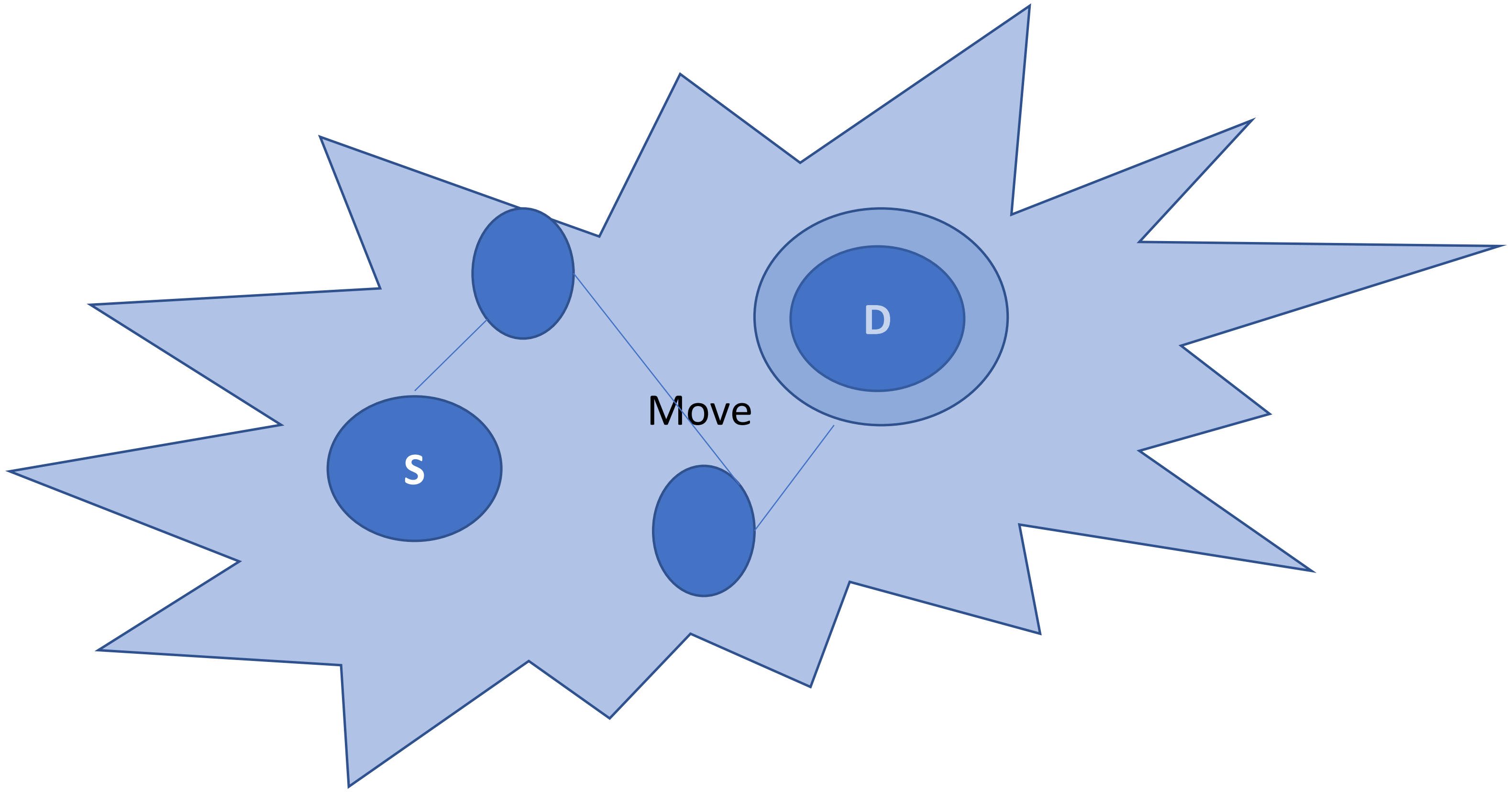


Search based method – state space



Search based method – state space

- Movegen(S) – find all possible neighbours

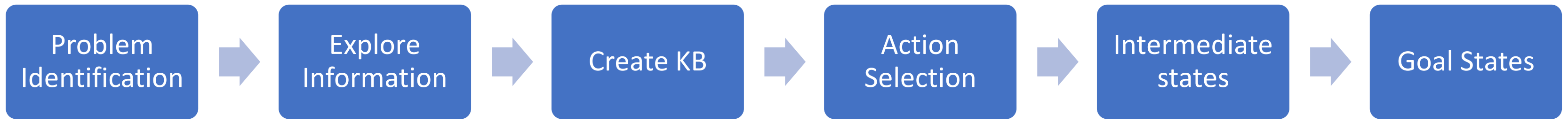


PROBLEM SOLVING

- Problem solving – area of finding answers for unknown situation
  - Understanding
  - Representation
  - Formulation
  - Solving
- Types,
  - Simple – Can be solved using deterministic approach
  - Complex – Lack of full information
- Humans?
  - Able to perceive, learn, use statistical methods, mathematical modelling to solve
  - AI do the same for the machine

PROBLEM SOLVING PROCESS

- Problem? – desired objective is not obvious
- Problem solving?
  - process of generating solution for given situation
  - Sequence of well defined methods that can handle doubts, inconsistency, uncertainty and ambiguity



Vacuum Cleaner – problem solving

Introduction

- well-known search problem for an agent which works on Artificial Intelligence
- our vacuum cleaner is our agent
- Goal – clean up the whole area

Understanding

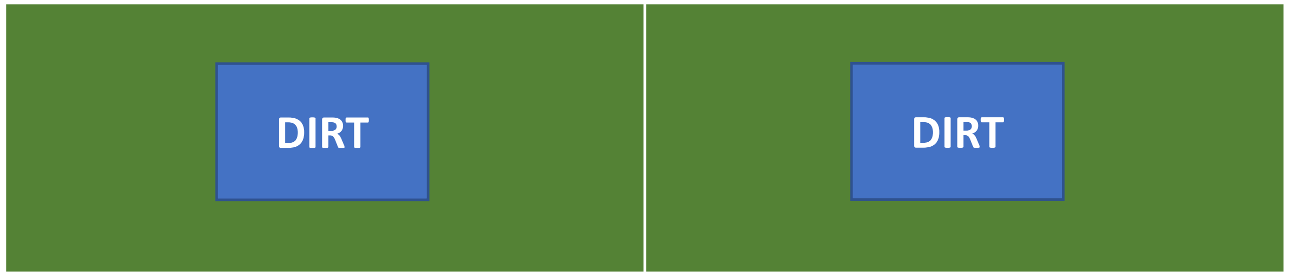
- Two rooms and one vacuum cleaner
- Dirt in both rooms – to be cleaned
- Vacuum cleaner present in any one room
- Goal – clean both rooms

Representation

- Two rooms

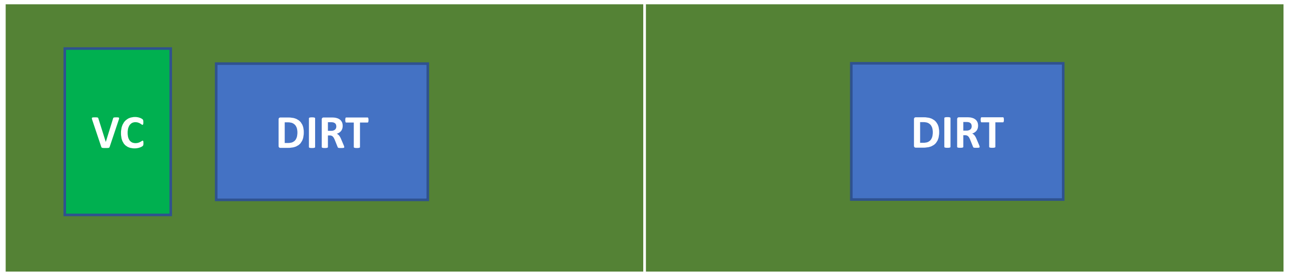
Representation

- Two rooms – with dirt



Representation

- Two rooms – with dirt

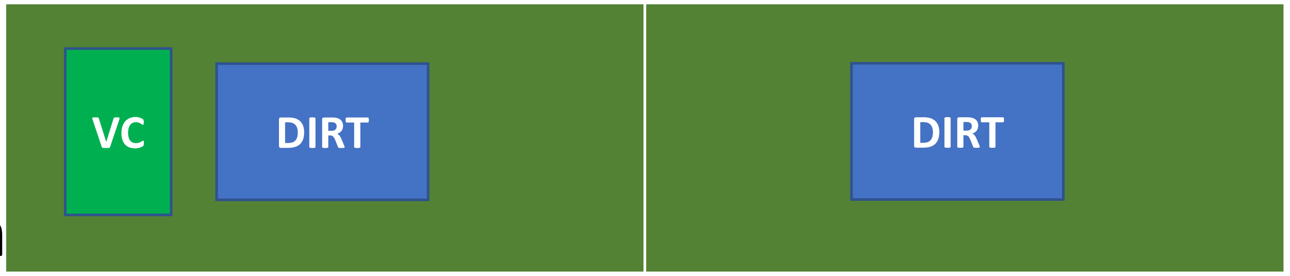


- Vacuum cleaner in any one room

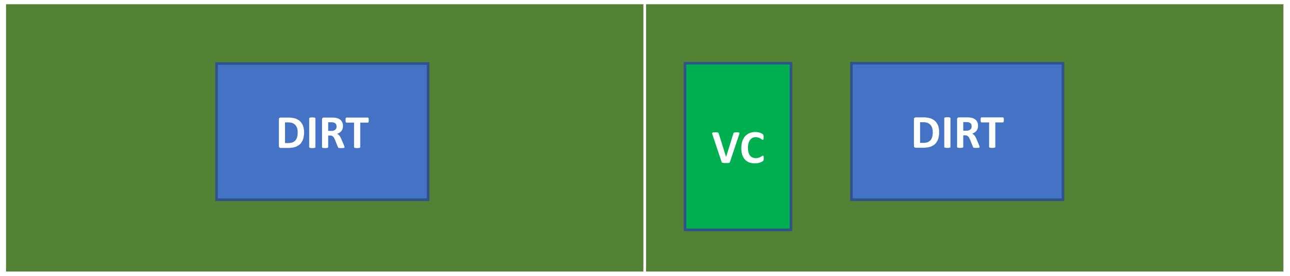
- **State representation**

8 possible states

1 – Dirt – both rooms – Vacuum cleaner – Left room

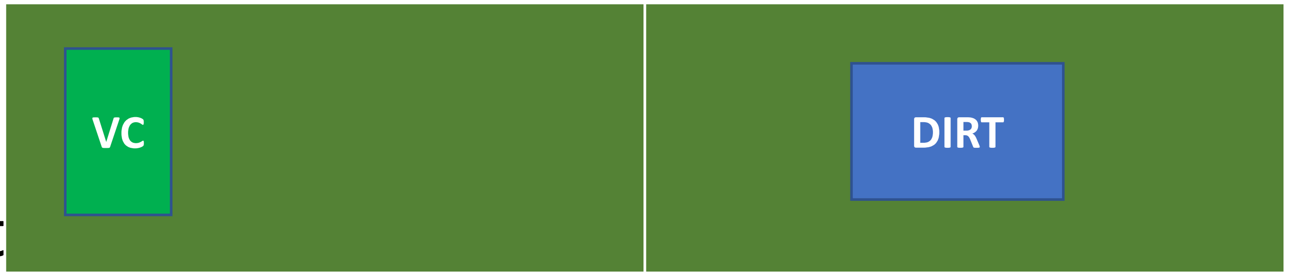


2 – Dirt – both

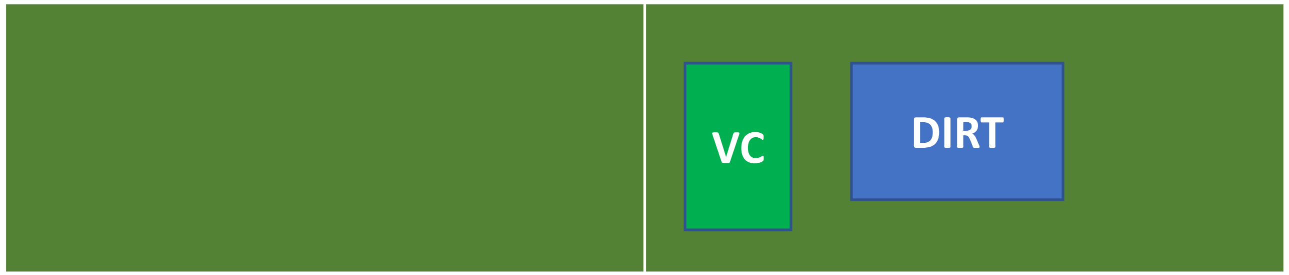


8 possible states

3 – Dirt - right room – Vacuum cleaner – Left room

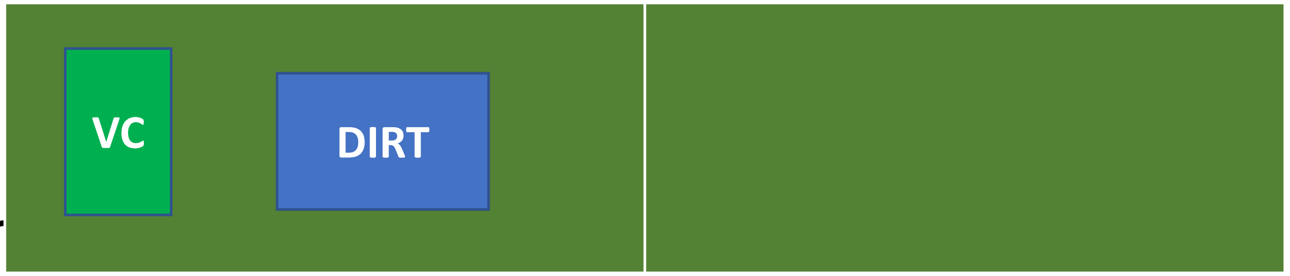


4 – Dirt – right

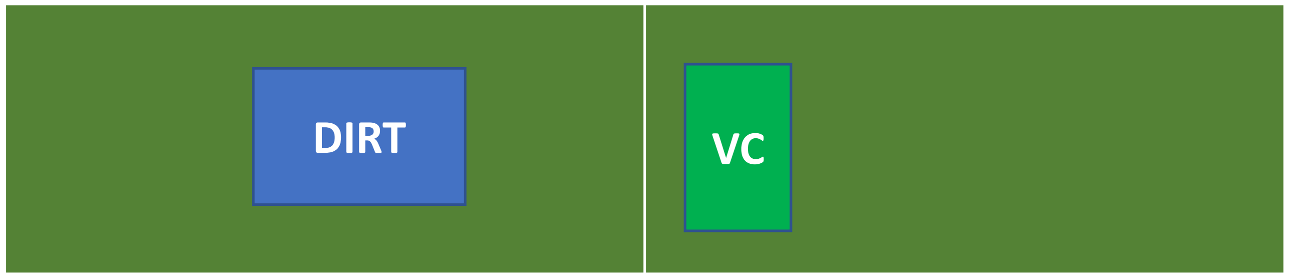


8 possible states

5 – Dirt – left room – Vacuum cleaner – Left room

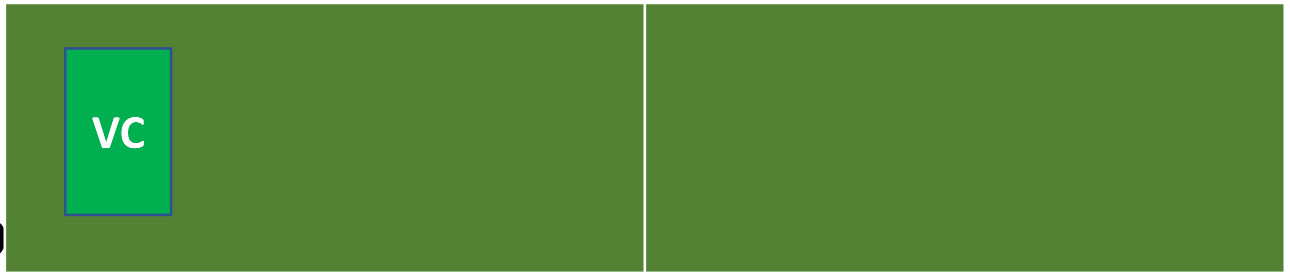


6 – Dirt – left r

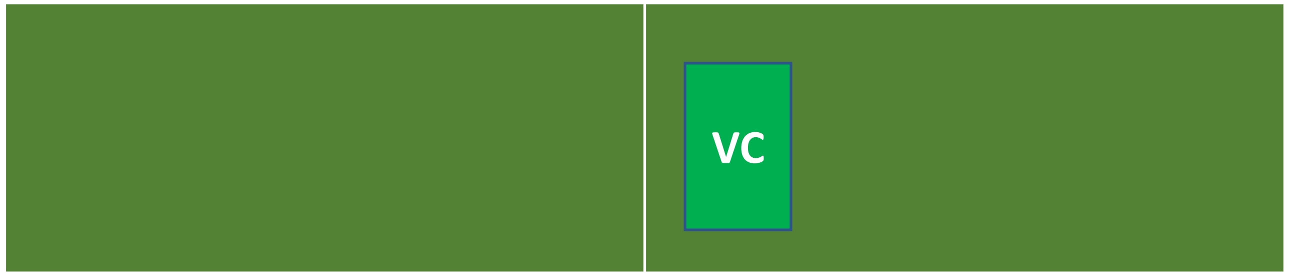


8 possible states

7 – No Dirt – both rooms – Vacuum cleaner – Left room



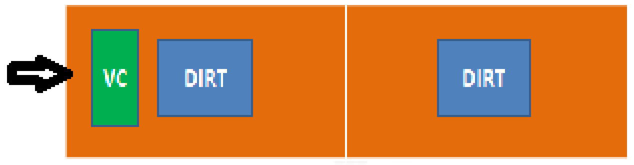
8 – No Dirt – b



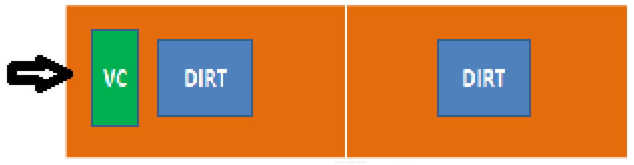
Formulation

- Possible action
    - Move Left
    - Move Right
    - Clean Dirt

Solving

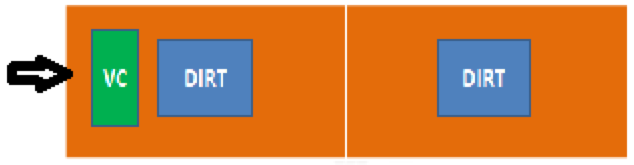


Solving



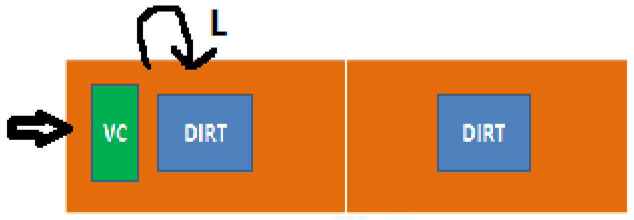
- –Move Left
- –Move Right
- –Clean Dirt

Solving



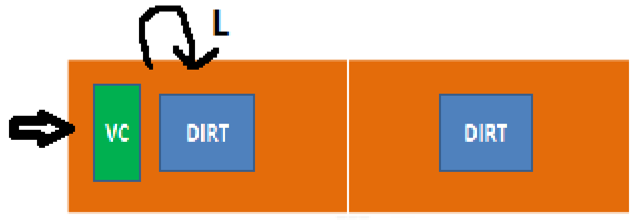
VC - Already Left
- Move Left
- Move Right
- Clean Dirt

Solving



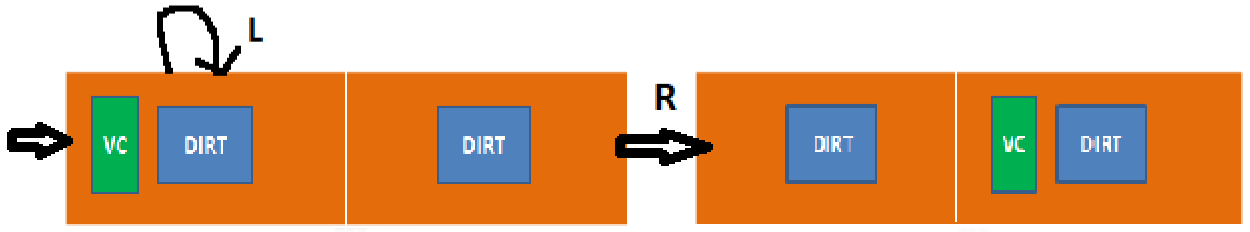
VC - Already Left
- Move Left
- Move Right
- Clean Dirt

Solving



- Move Right
- Clean Dirt

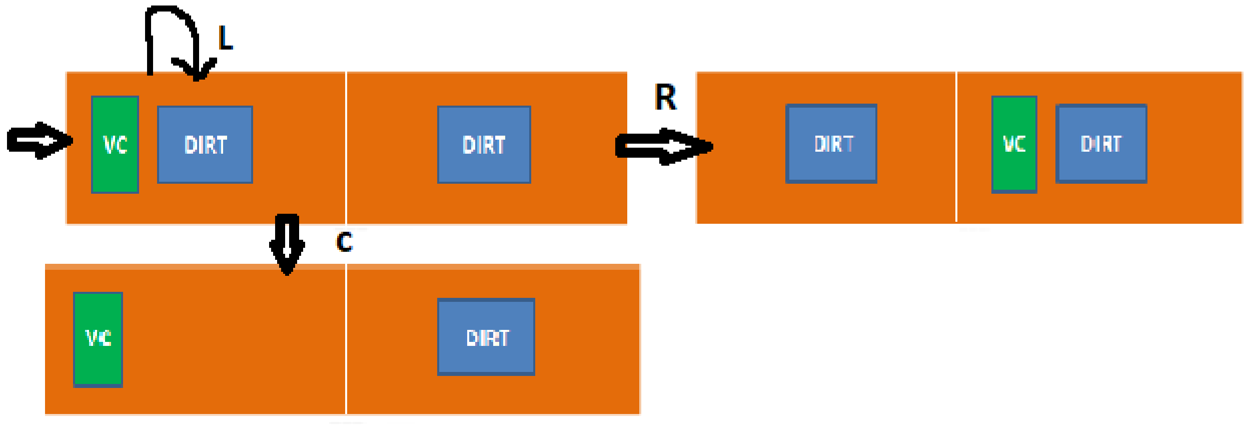
Solving



–Move Right

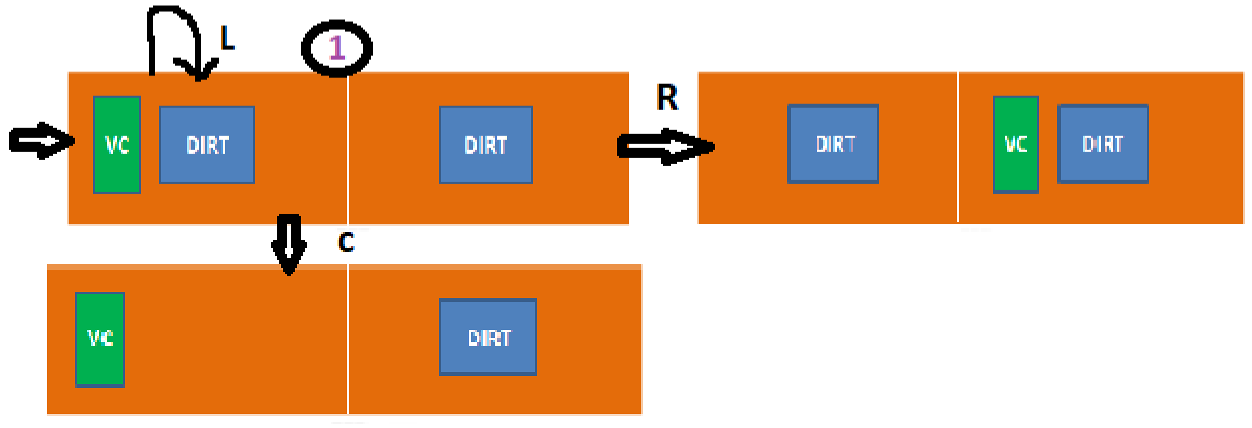
–Clean Dirt

Solving

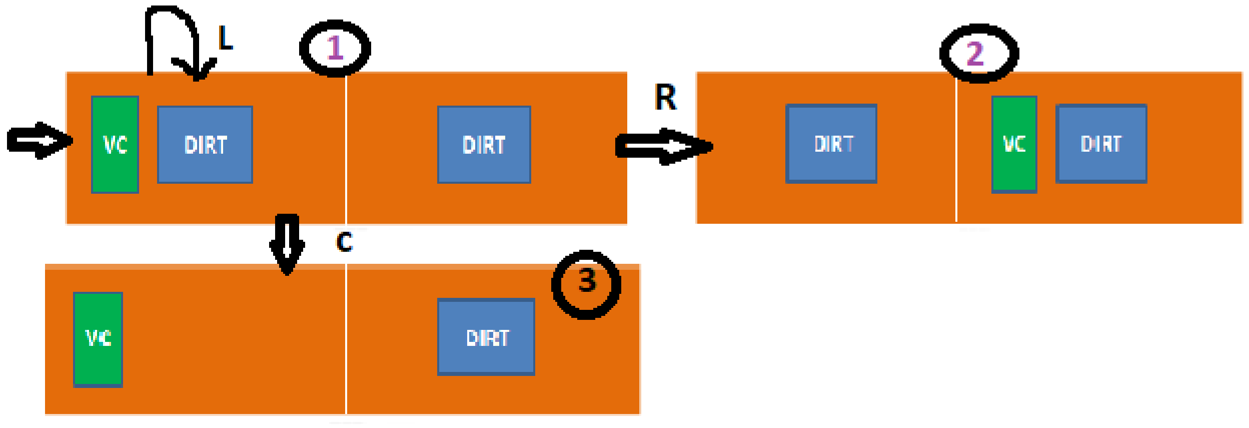


–Clean Dirt

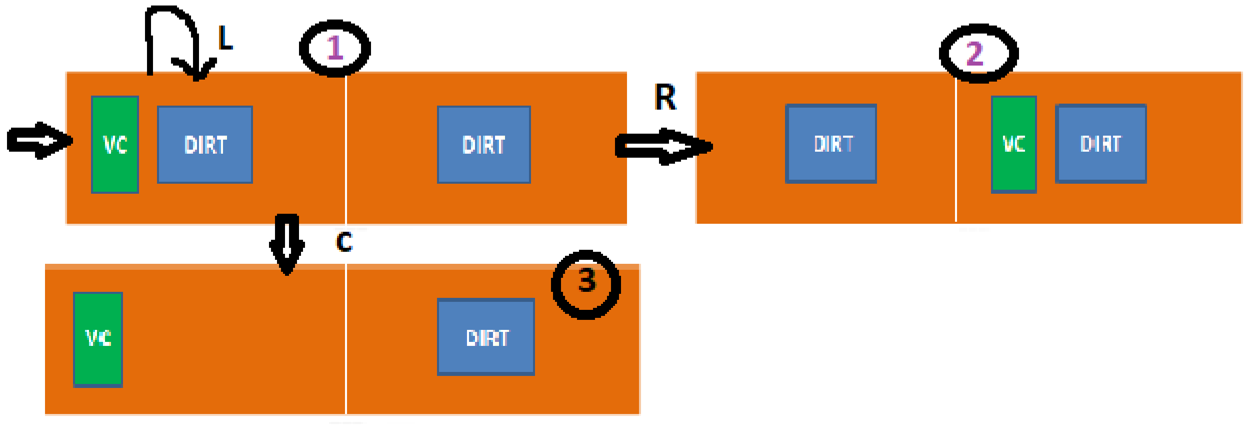
Solving



Solving



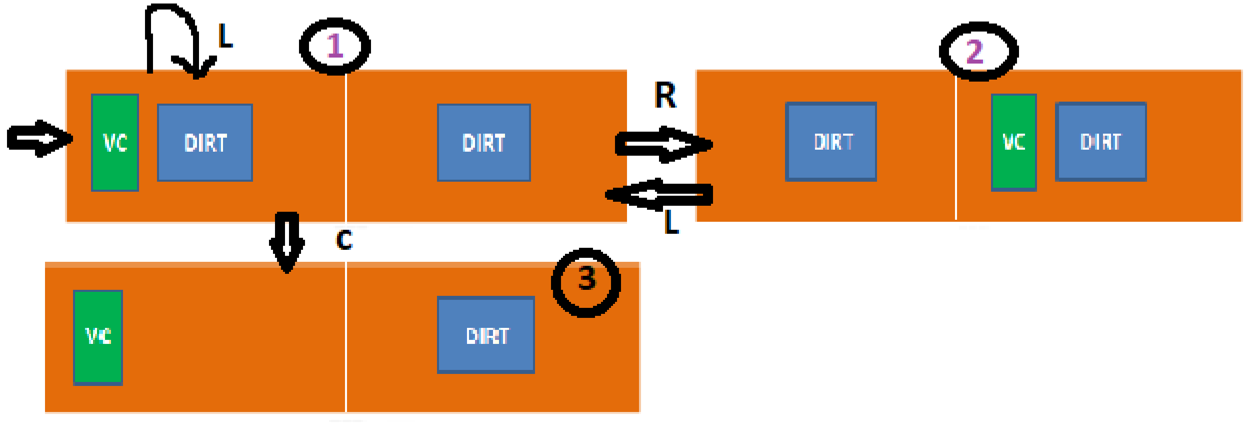
Solving



State 2

- **move left**
- **move right**
- **clean dirt**

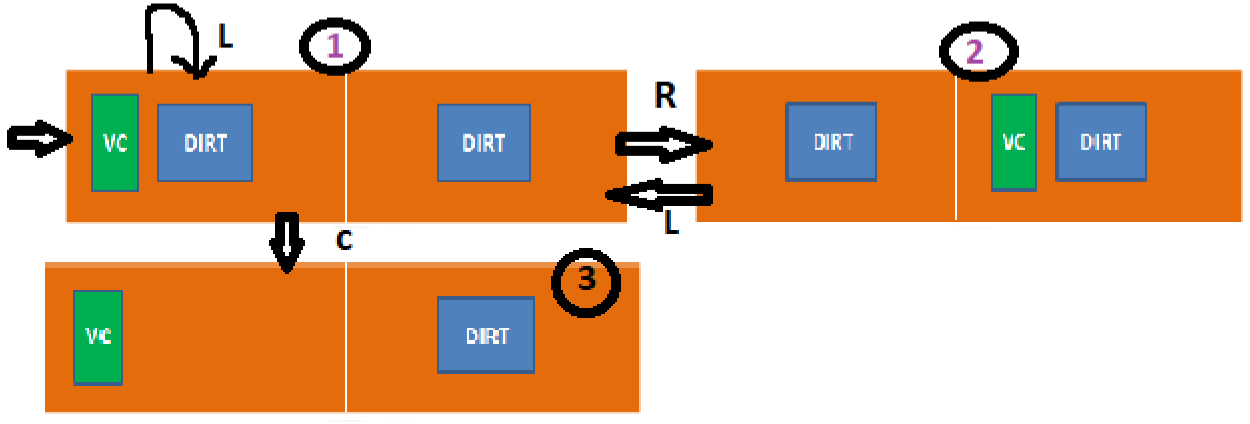
Solving



State 2

- $\boxed{\text{move left}}$
- move right
- clean dirt

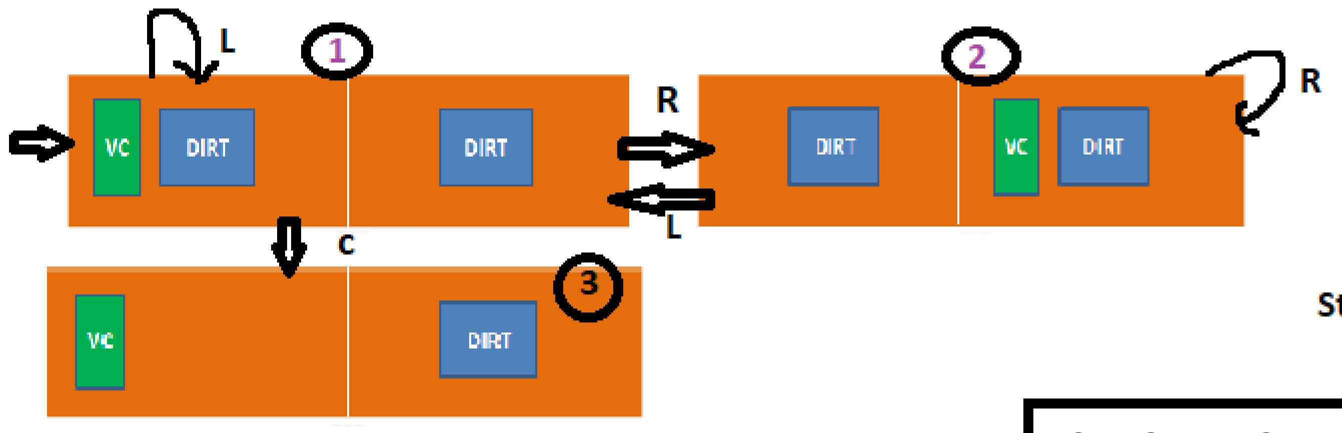
Solving



State 2

Already in right - move right
- clean dirt

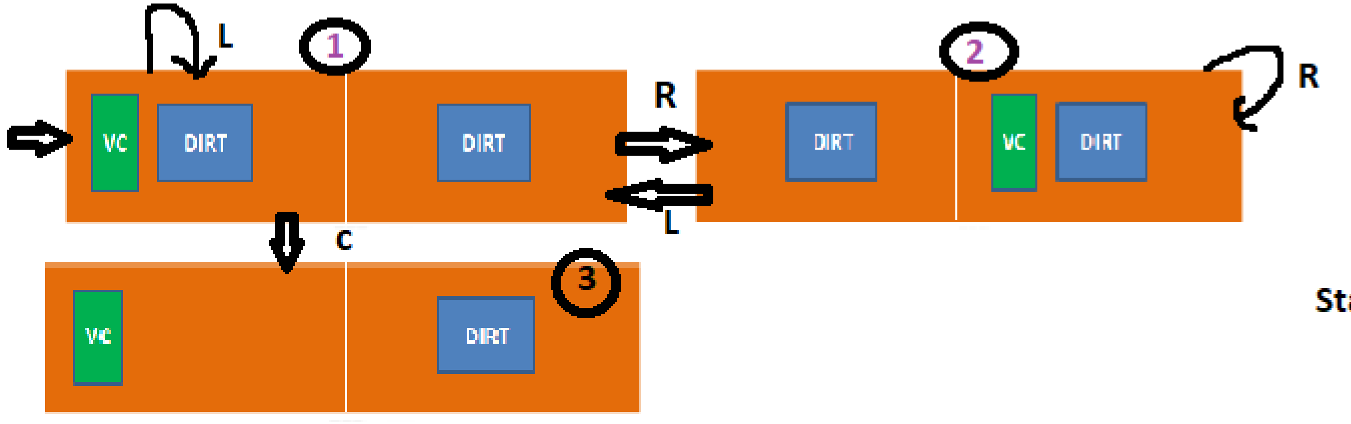
Solving



State 2

Already in right - move right
- clean dirt

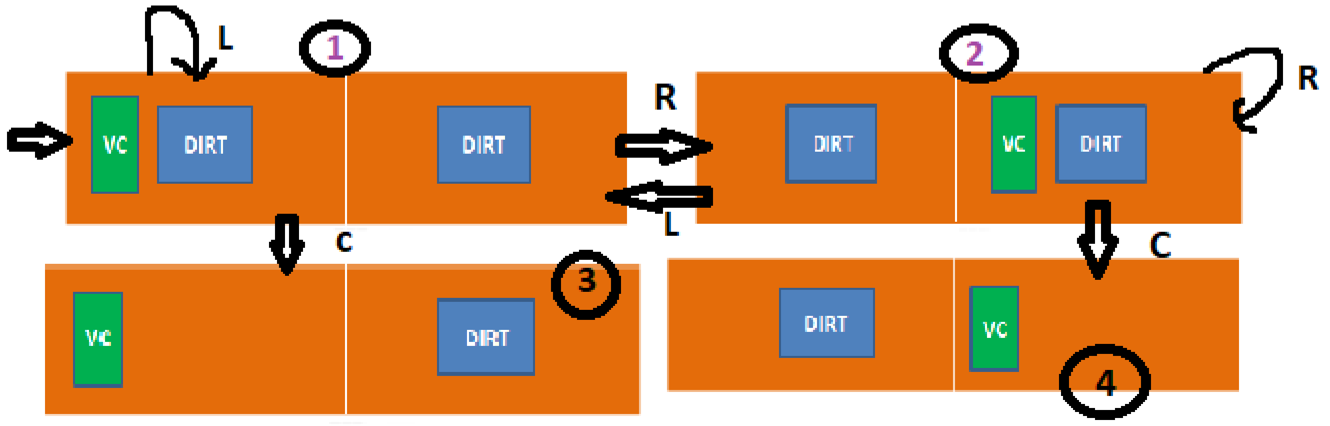
Solving



State 2

- clean dirt

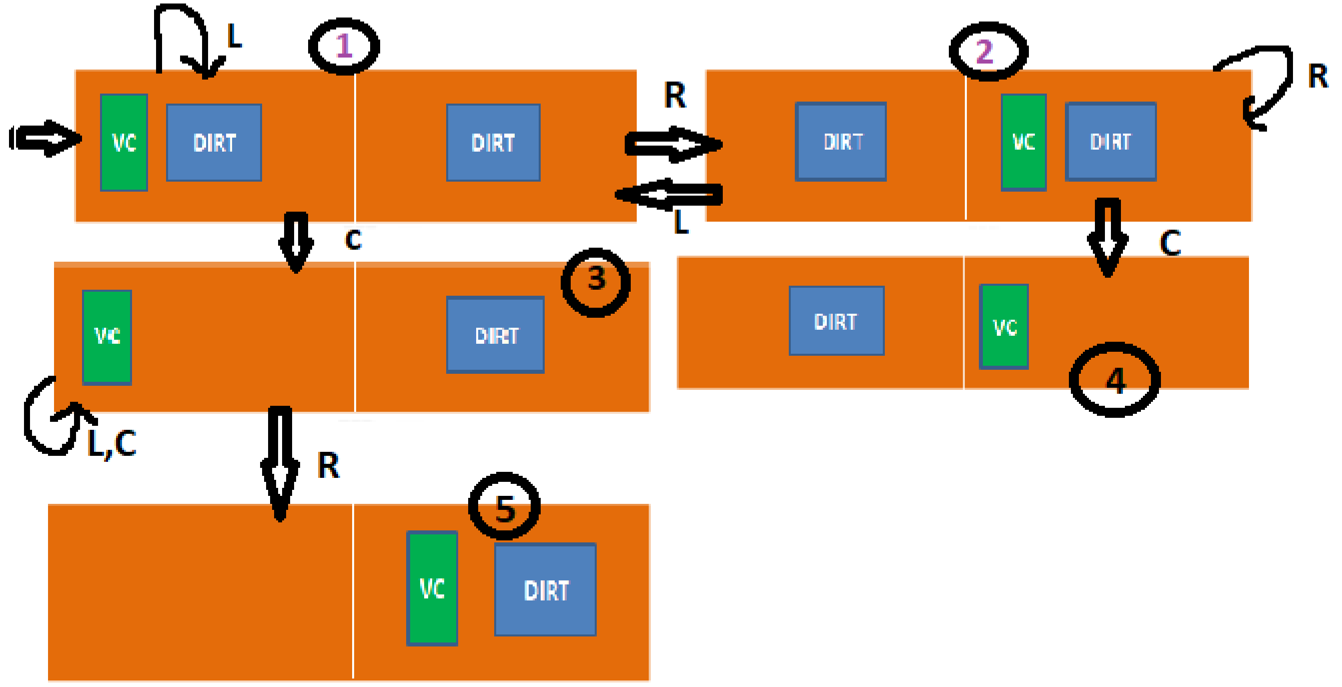
Solving



State 2

Clean dirt

Solving



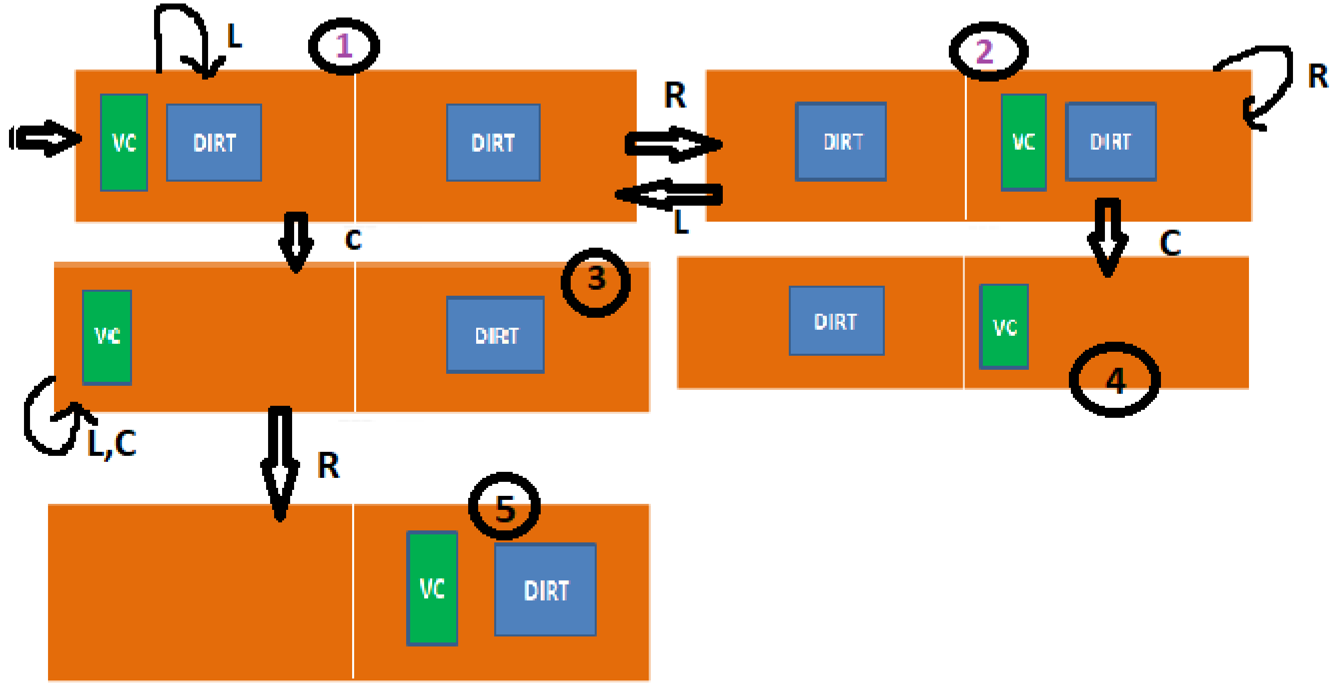
State 3

Move left - 3

Move right - 5

Clean Dirt - 3

Solving



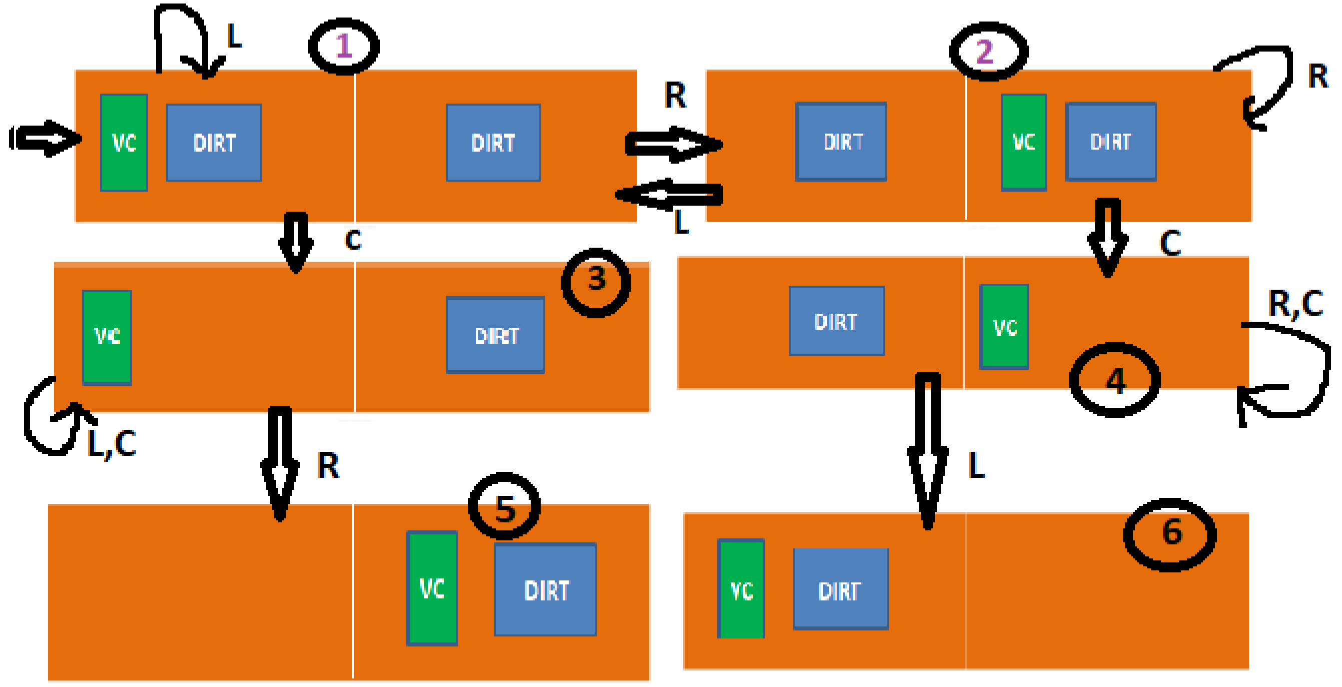
State 4

Move left - 6

Move right - 4

Clean Dirt - 4

Solving



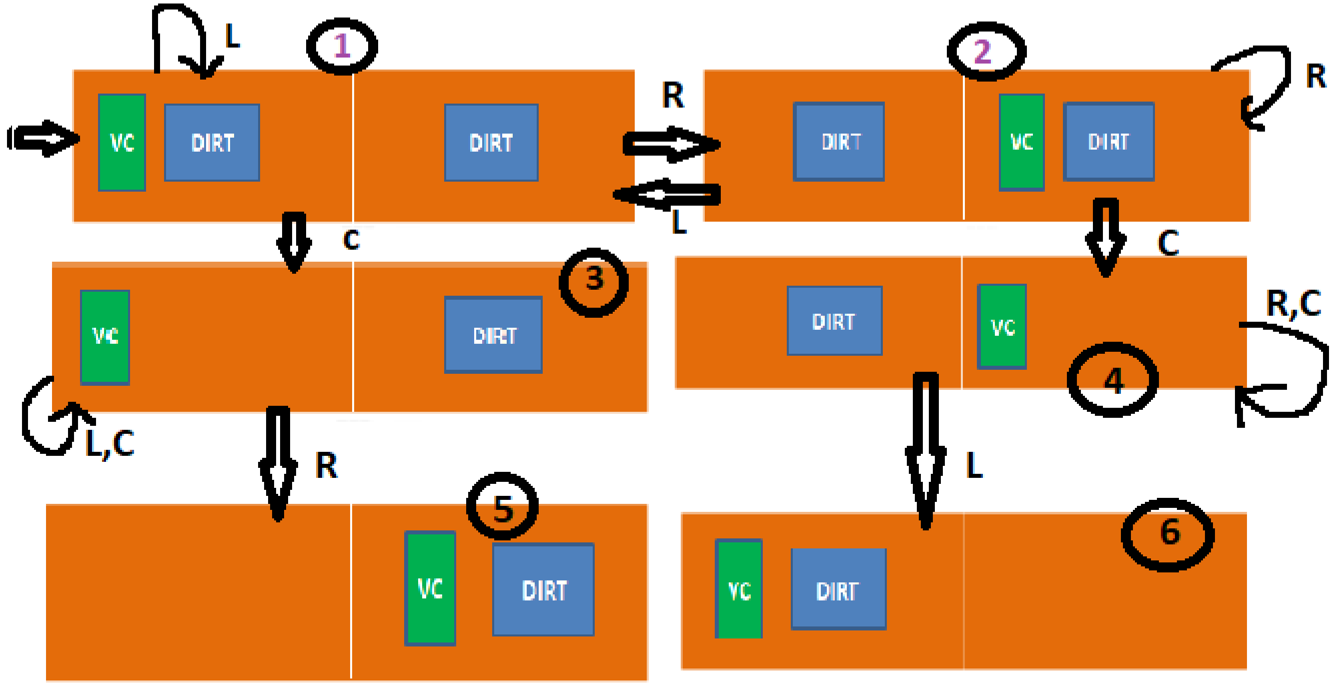
State 4

Move left - 6

Move right - 4

Clean Dirt - 4

Solving



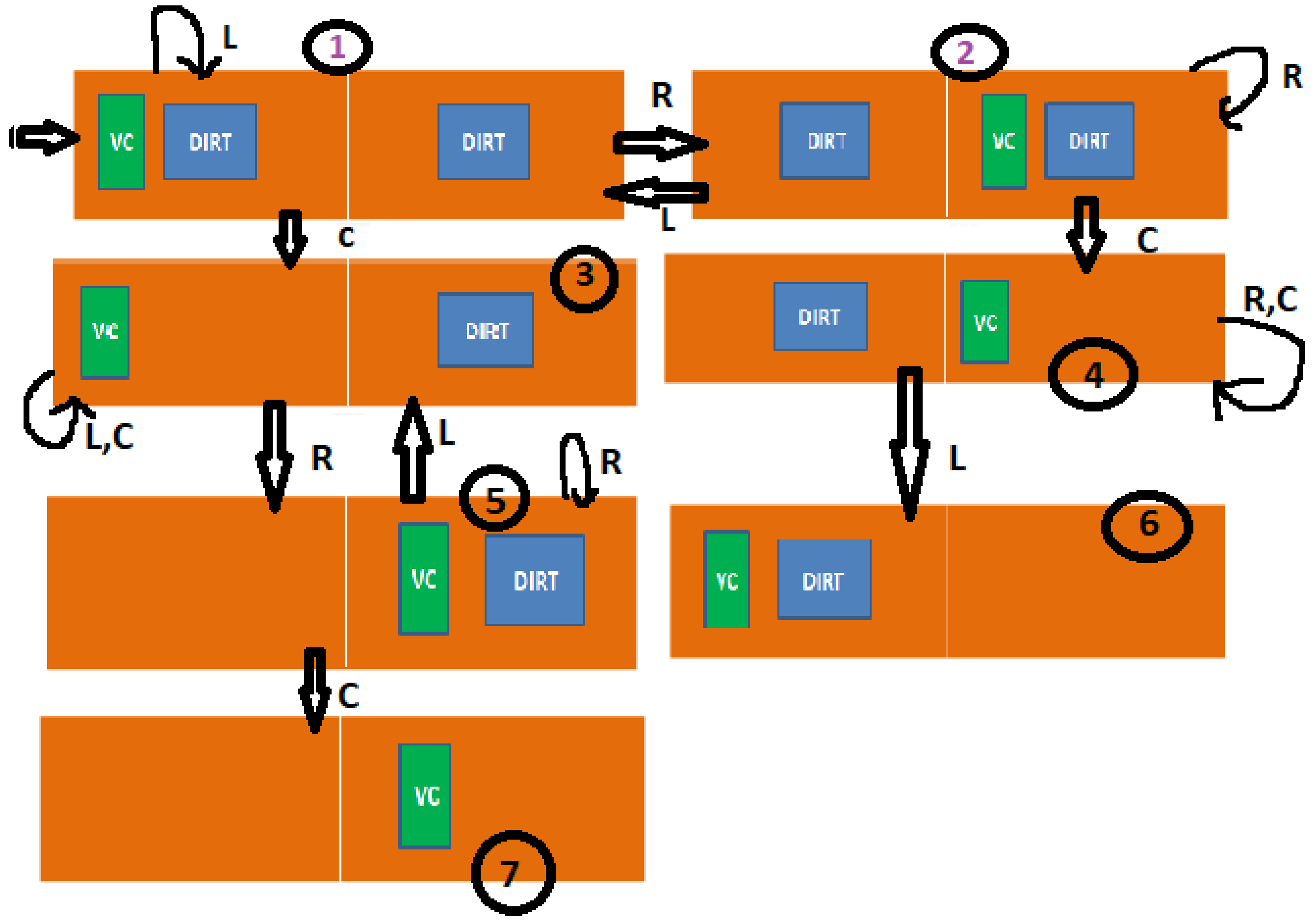
State 5

Move left - 3

Move right - 5

Clean Dirt - 7

Solving



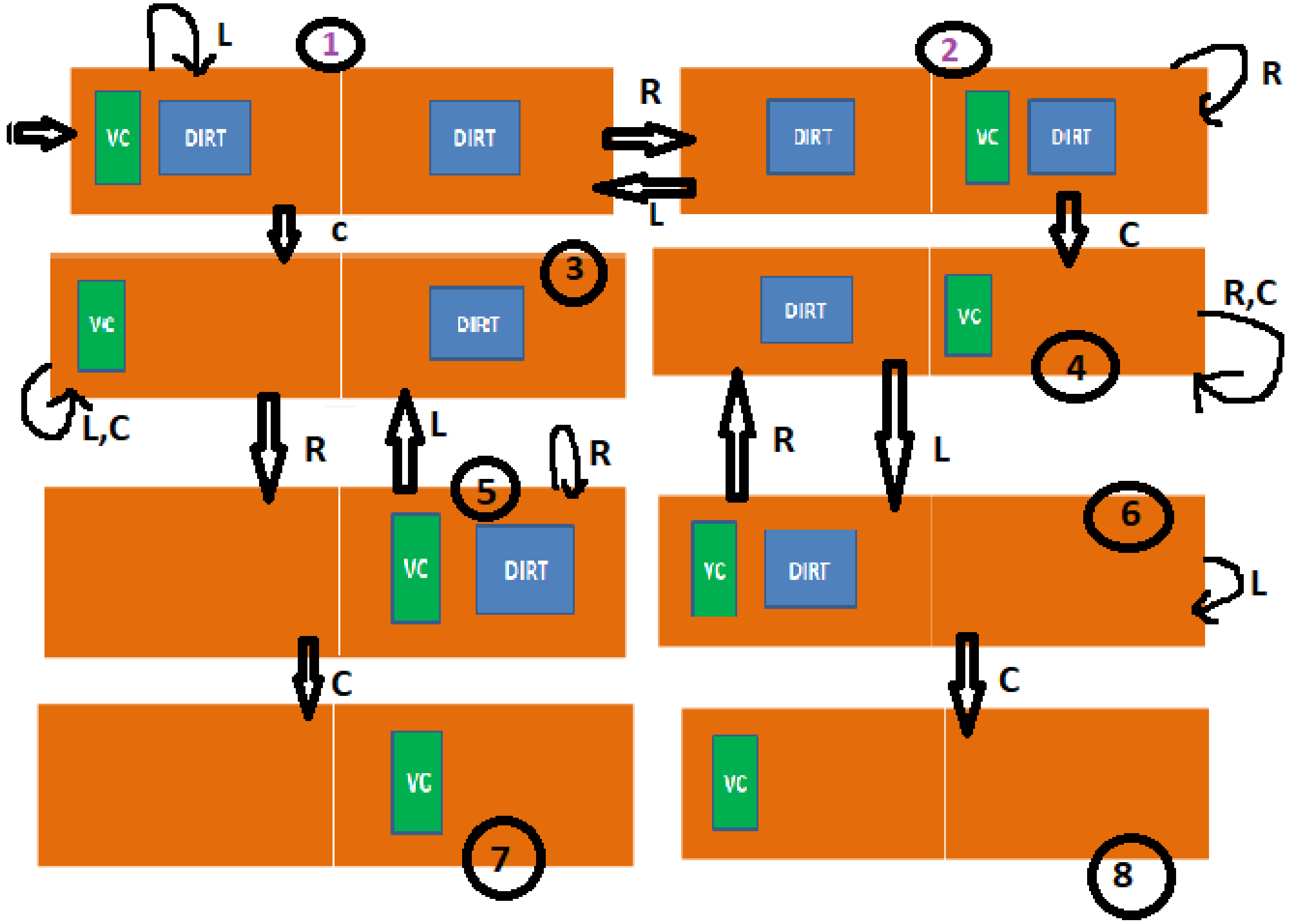
State 5

Move left - 3

Move right - 5

Clean Dirt - 7

Solving



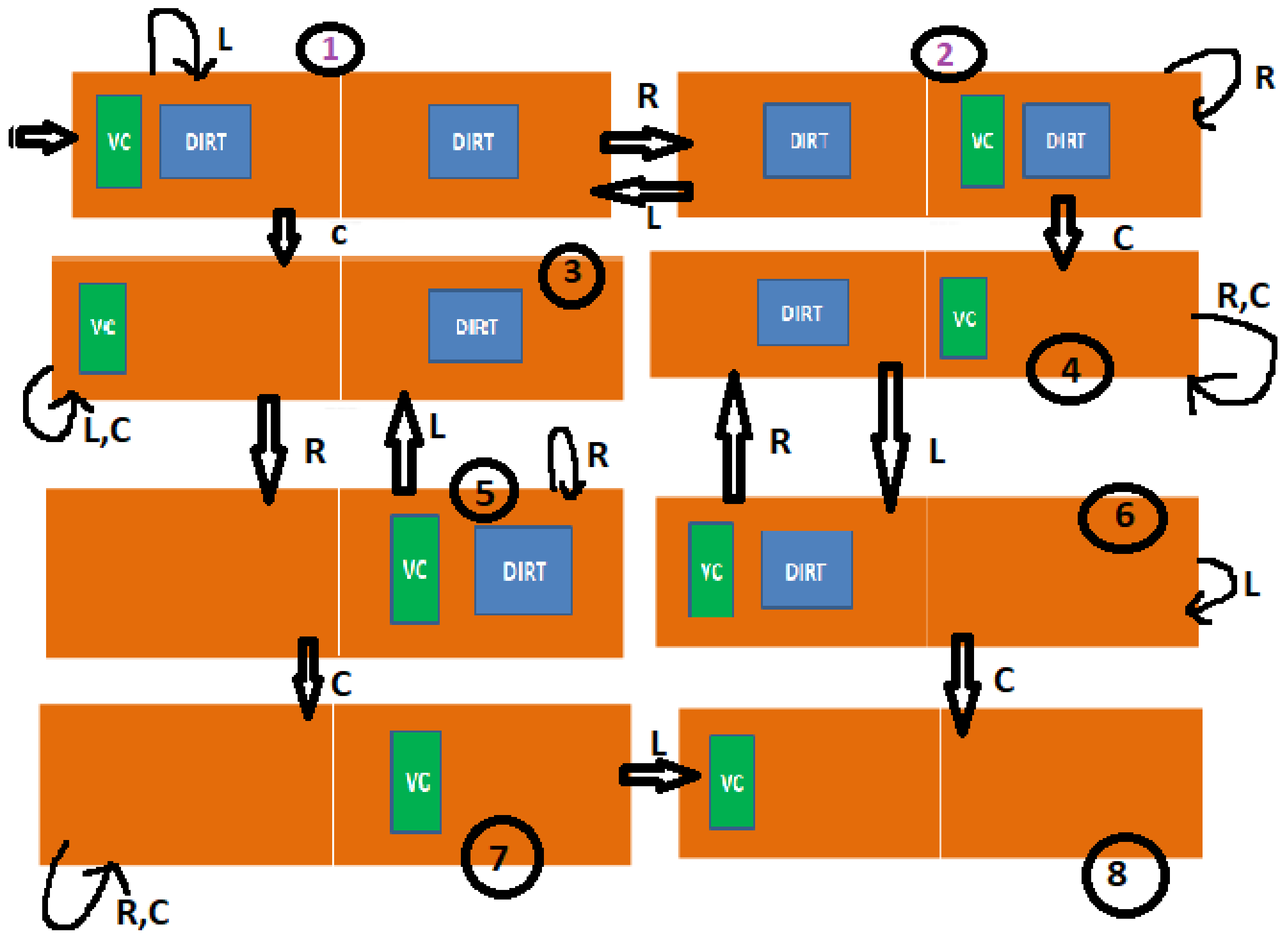
State 6

Move left - 6

Move right - 4

Clean Dirt - 8

Solving



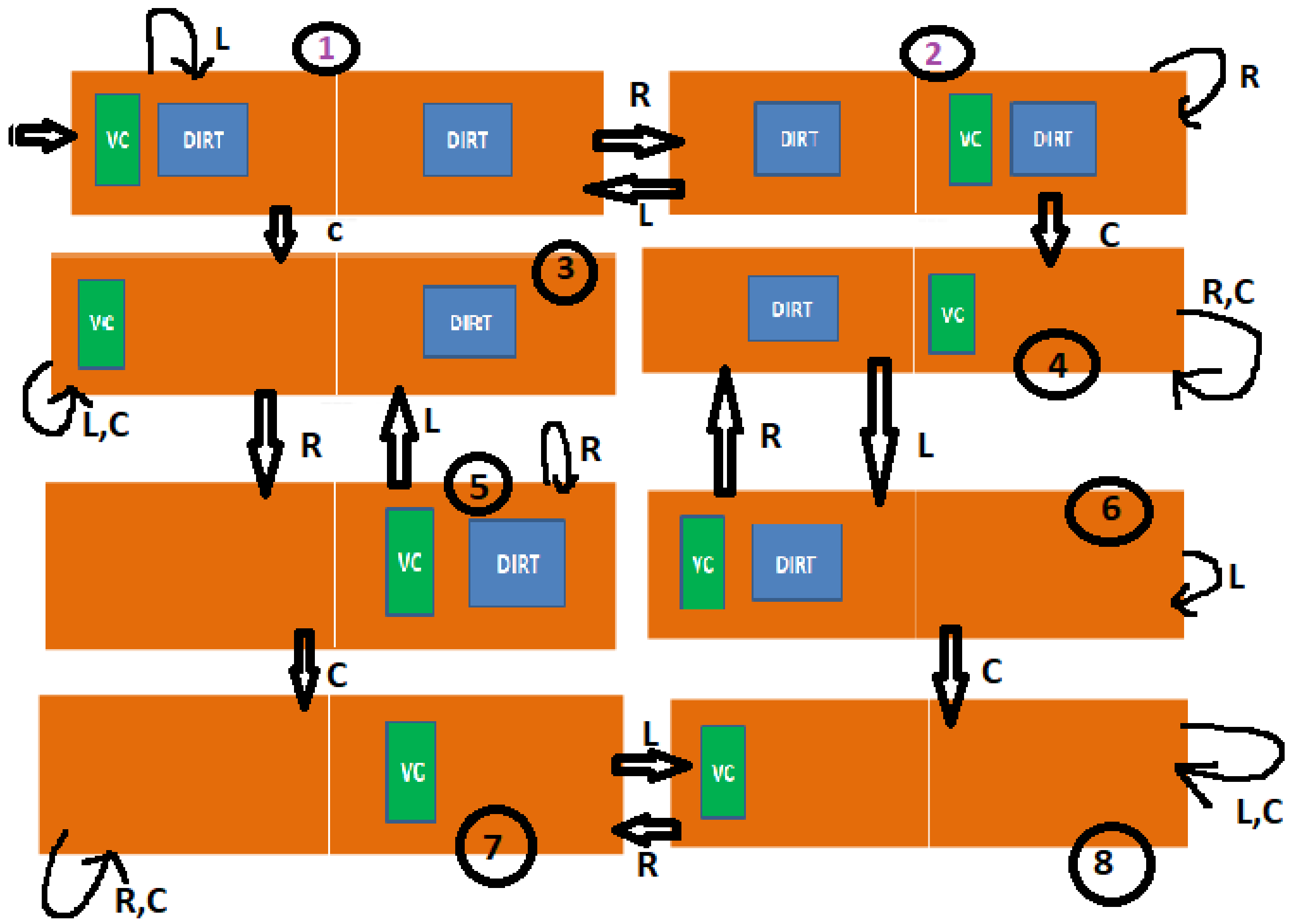
State 7

Move left - 8

Move right - 7

Clean Dirt - 7

Solving



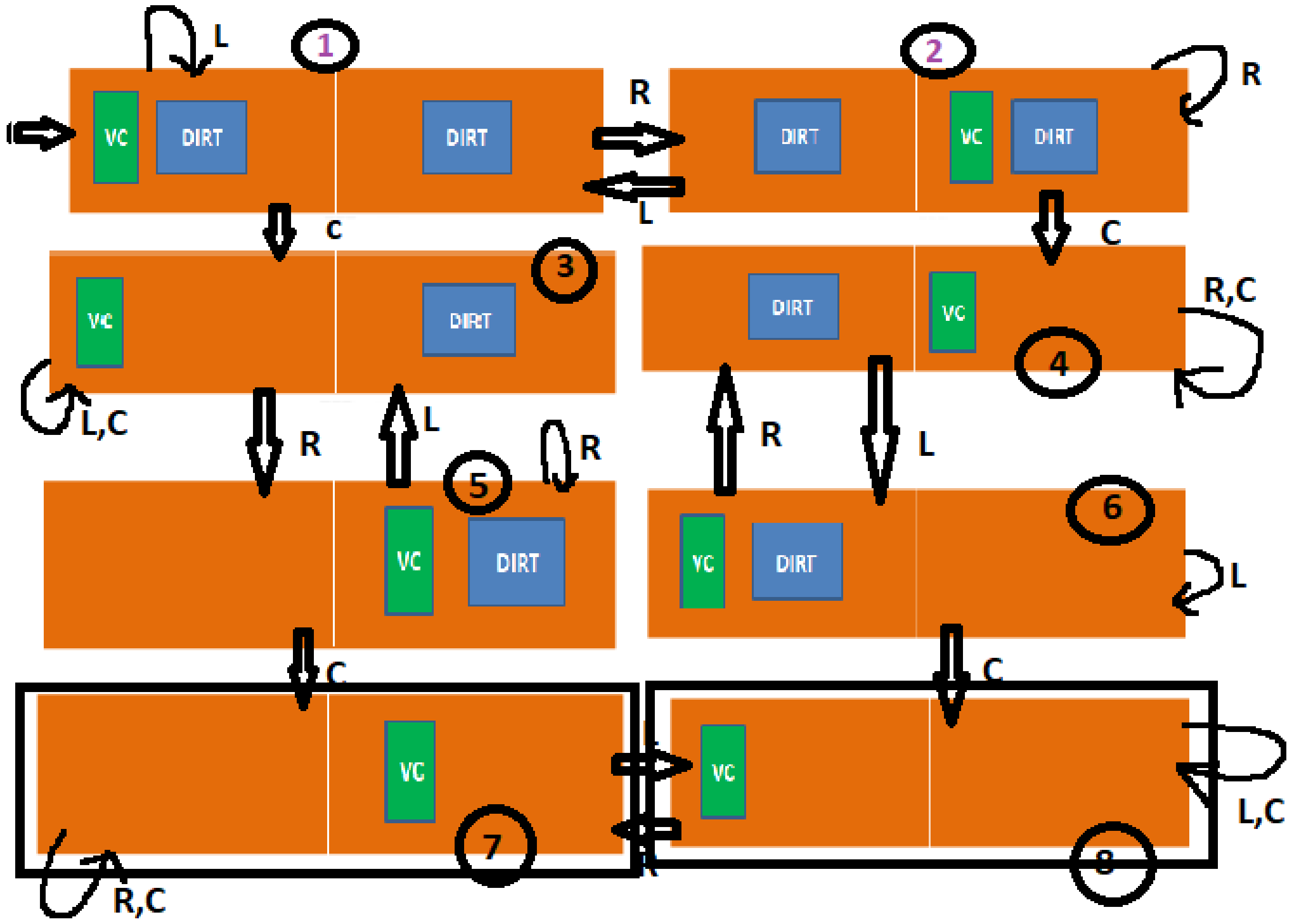
State 8

Move left - 8

Move right - 7

Clean Dirt - 8

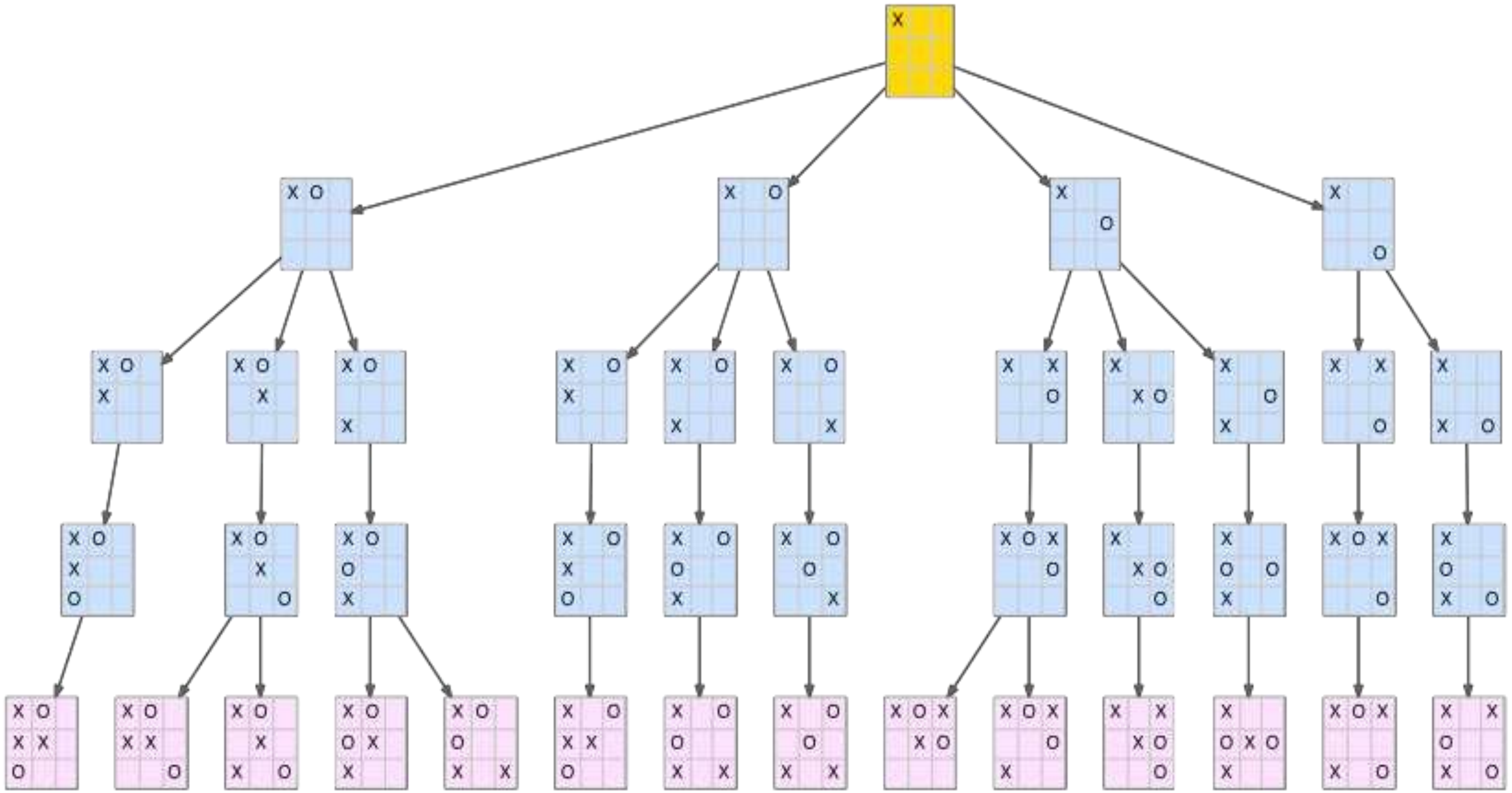
Solving



Problem

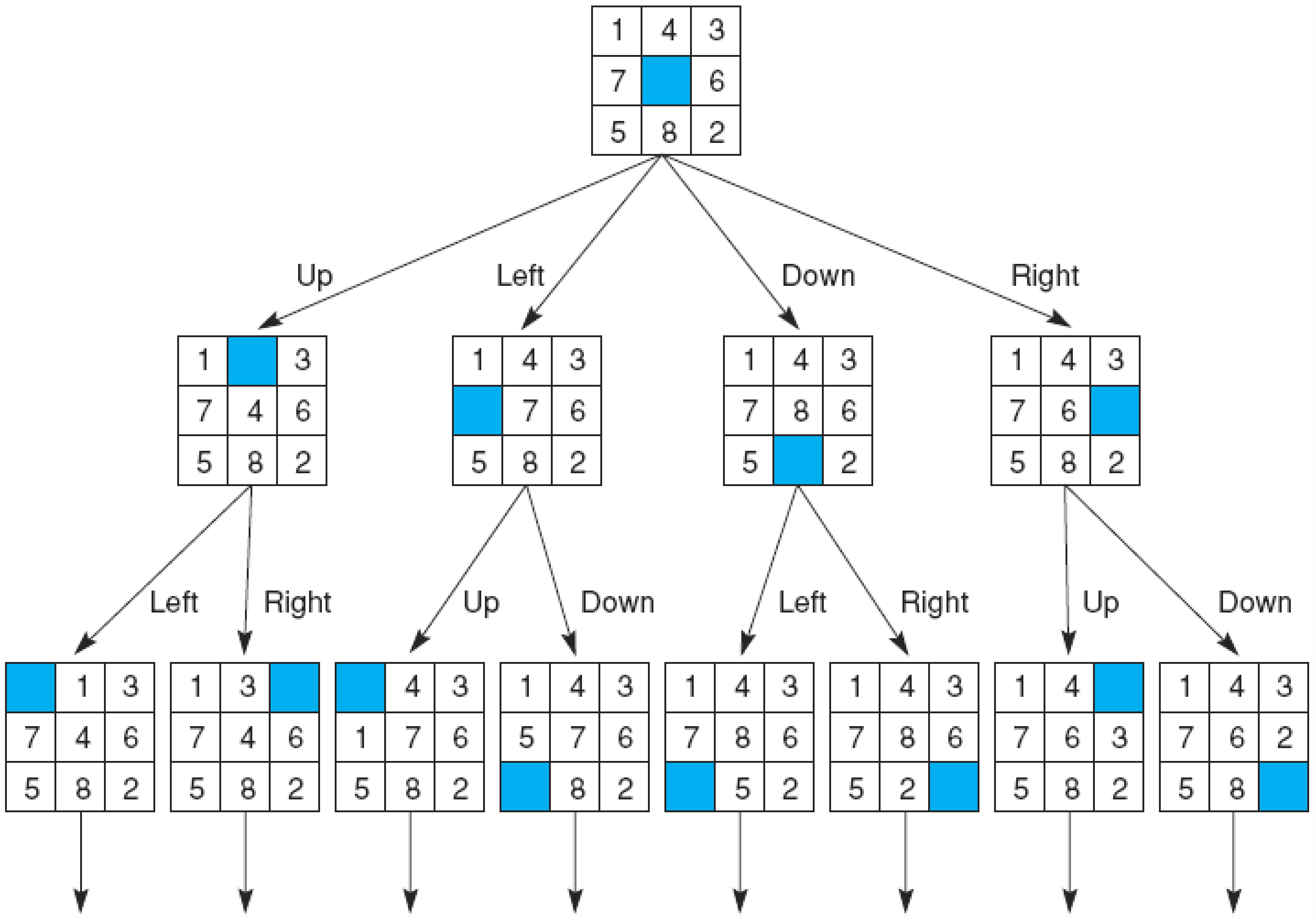
- In AI, formally define a problem as
  - a space of all possible configurations where each configuration is called a state
  - The ***state-space*** is the configuration of the possible states and how they connect to each other e.g. the legal moves between states.
  - an initial state
  - one or more goal states
  - a set of rules/operators which move the problem from one state to the next
- In some cases, we may enumerate all possible states
  - but usually, such an enumeration will be overwhelmingly large so we only generate a portion of the state space, the portion we are currently examining
  - we need to search the state-space to find an optimal path from a start state to a goal state.

State space: Tic-Tac-Toe



Goal: Arrange in horizontal or vertical or diagonal to win

State space: 8 Puzzle



The 8 puzzle search space consists of 8! states (40320)

Categories of problems

Structured problems –goal state defined

Unstructured problems- goal state not known

Linear problems- based on dependent variable

Non linear problems- no dependency between variables

Problem Types

1. Deterministic or observable (single-state)
2. Non-observable (multiple-state)
3. Non-deterministic or partially observable
4. Unknown state space

Problem Types

1. Deterministic or observable(Single-state problems)

- Each state is fully observable and it goes to one definite state after any action.
- Here , the goal state is reachable in one single action or sequence of actions.
- Deterministic environments ignore uncertainty.
- *Ex- Vacuum cleaner with sensor.*

Problem Types

2. Non-observable(Muliple-state problems) / conformant problems

- **Problem – solving agent does not have any information about the state.**
- **Solution may or may not be reached.**
- *Ex- In case of vacuum cleaner , the goal state is to clean the floor rather clean floor. Action is to suck if there is dirt. So , in non-observable condition , as there is no sensor , it will have to suck the dirt , irrespective of whether it is towards right or left . Here , the solution space is the states specifying its movement across the floor.*

Problem Types

3. Non-deterministic(partially observable) / contingency problem

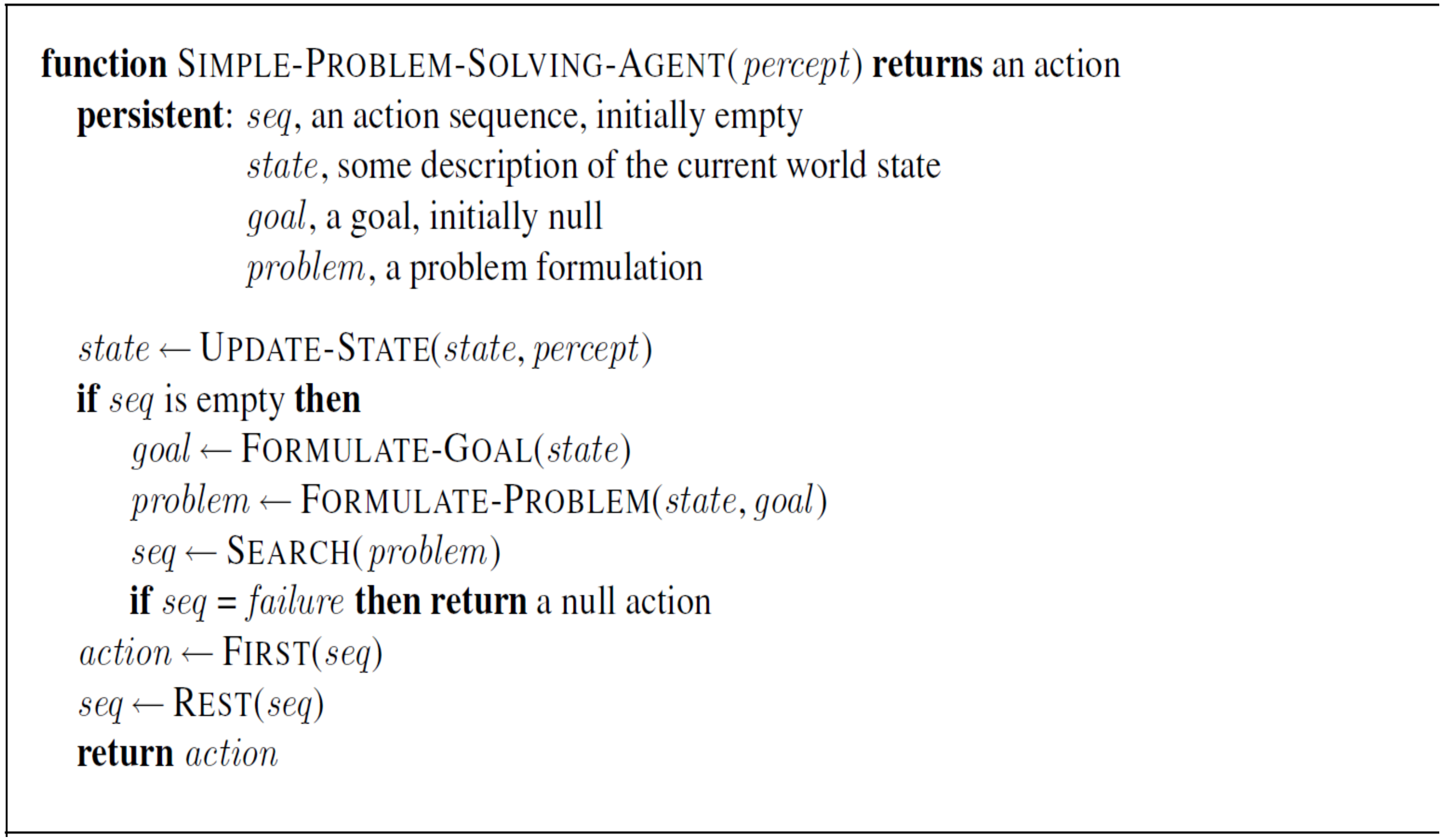
- **The effect of action is not clear.**
- **Percepts provide new information about the current state.**
- *Ex- If we take Vacuum cleaner , and now assume that the sensor is attached to it , then it will suck if there is dirt. Movement of the cleaner will be based on its current percept.*

Problem Types

4. Unknown state space problems

- **Typically exploration problems**
- **States and impact of actions are not known**
- ***Ex- online search that involves acting without compete knowledge of the next state or scheduling without map.***

Problem Solving with AI
“ Formulate , Search , Execute “ design for agent



```
function SIMPLE-PROBLEM-SOLVING-AGENT($percept$) returns an action
  persistent: $seq$, an action sequence, initially empty
              $state$, some description of the current world state
              $goal$, a goal, initially null
              $problem$, a problem formulation

  $state \leftarrow$ UPDATE-STATE($state, percept$)
  if $seq$ is empty then
     $goal \leftarrow$ FORMULATE-GOAL($state$)
     $problem \leftarrow$ FORMULATE-PROBLEM($state, goal$)
     $seq \leftarrow$ SEARCH($problem$)
     if $seq = failure$ then return a null action
  $action \leftarrow$ FIRST($seq$)
  $seq \leftarrow$ REST($seq$)
  return $action$
```

A simple problem-solving agent. It first formulates a goal and a problem, searches for a sequence of actions that would solve the problem, and then executes the actions one at a time. When this is complete, it formulates another goal and starts over.

Searching for solution

- Searching for AI problems involves performing an action to go to one proper state among possible numerous states of agents

- Thus the process of finding solution can be searching of the best among all the possible states

Characteristics of AI

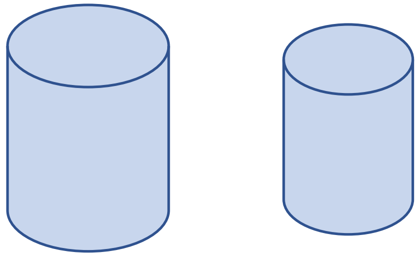
- AI problems have large number of combination of solution
- AI program manipulate large number of Symbolic information
- AI problem uses heuristic search to cope with the combination of exploration of solution
- Ability to learn
- AI problems can be solved with or without the use of AI technique

Water Jug problem

- A Water Jug Problem:

You are given two jugs, a 4-gallon one and a 3-gallon one, a pump which has unlimited water which you can use to fill the jug, and the ground on which water may be poured. Neither jug has any measuring markings on it. How can you get exactly 2 gallons of water in 4-gallon jug?

(x,y)	(x,y)
(0,0)	(4,0)
(0,3)	(1,3)
(3,0)	(1,0)
(3,3)	(0,1)
(4,2)	(4,1)
	(2,3)



With out AI

- First pour 3l water in 2nd jug
  - X=0,y=3
- Pour 3l water in 2nd jug to x
  - X=3,y=0
- Pour the extra water from outside in 2nd jug
  - X=3,y=3
- The total capacity of jug 1 is 4 ,so fill the remaining in jug 1 So
  - x=4,y=2
- Here 2l water is filled in jug 2

Representation

- State Representation and Initial State
- we will represent a state of the problem as a
- tuple (x, y) where x represents the amount of water in the 4-gallon jug and y represents the amount of water in the 3-gallon jug.
- Note $0 \leq x \leq 4$, and $0 \leq y \leq 3$. Our initial state: (0,0)

Production rules - Formulation

1. Fill 4-gal jug $(x,y) \rightarrow (4,y)$ where ,$x < 4$

2. Fill 3-gal jug $(x,y) \rightarrow (x,3)$ where, $y < 3$

Production rules - Formulation

3. Empty 4-gal jug on ground $(x,y) \rightarrow (0,y)$ where, $x > 0$

4. Empty 3-gal jug on ground $(x,y) \rightarrow (x,0)$ where, $y > 0$

Production rules - Formulation

5.Empty some water in 3 gallon jug$(x,y) \rightarrow (x,y-d)$ where, $y>0$

6. Empty some water in 4 gallon jug$(x,y) \rightarrow (x-d,y)$ where, $x>0$

Production rules - Formulation

7. Pour water from 3-gal jug $(x,y) \rightarrow (4, y - (4 - x))$ to fill 4-gal jug $x+y \geq 4$ and $y > 0$

8. Pour water from 4-gal jug $(x,y) \rightarrow (x - (3-y), 3)$ to fill 3-gal-jug $x+y \geq 3$ and $x > 0$

Production rules - Formulation

9. Pour all of water from 3-gal jug $(x,y) \rightarrow (x+y, 0)$ into 4-gal jug $0 < x+y \leq 4$ and $y \geq 0$

10. Pour all of water from 4-gal jug $(x,y) \rightarrow (0, x+y)$ into 3-gal jug $0 < x+y \leq 3$ and $x \geq 0$

One solution

Gals in 4-gal jug	Gals in 3-gal jug	Rule Applied
0	0	
		1. Fill 4
4	0	
		8. Pour 4 into 3 to fill
1	3	
		4. Empty 3
1	0	
		10. Pour all of 4 into 3
0	1	
		1. Fill 4
4	1	
		8. Pour into 3
2	3	

Problem types

- single-state problem-Agent knows exactly what each of its actions does and it can calculate exactly which state it will be in after any sequence of actions.
- multiple-state problem-when the world is not fully accessible, the agent must reason about sets of states that it might get to, rather than single states.
- contingency problem-the agent may be in need to now calculate a whole tree of actions, rather than a single action sequence in which each branch of the tree deals with a possible contingency that might arise.
- exploration problem-the agent learns a "map“ of the environment, which it can then use to solve subsequent problems.

Problem Characteristics

- To choose the most appropriate method
- Its necessary to analyse the problem

Problem Characteristics

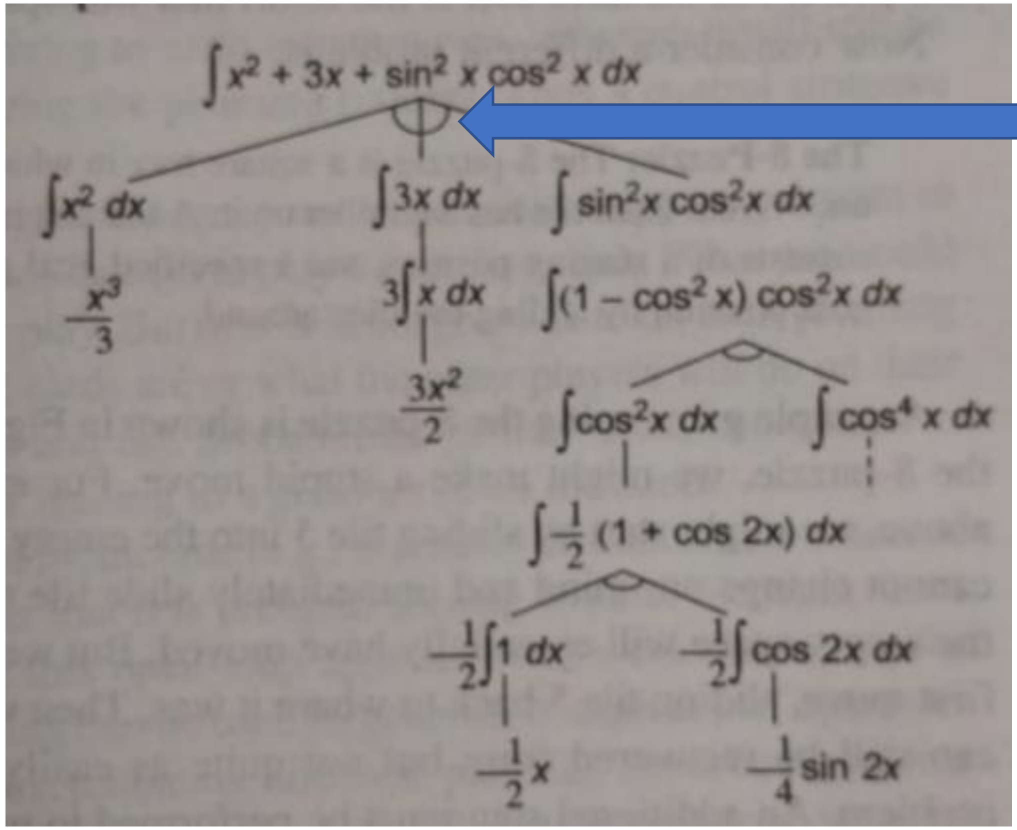
1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Is the problem Decomposable?

- Decomposable problem:

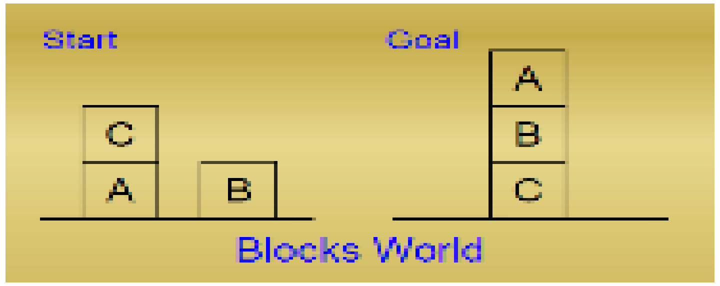


$\int x^2 + 3x + \sin^2 x \cos^2 x \, dx$
$\int x^2 \, dx$
$\frac{x^3}{3}$
$\int 3x \, dx$
$3\int x \, dx$
$\frac{3x^2}{2}$
$\int \sin^2 x \cos^2 x \, dx$
$\int (1 - \cos^2 x) \cos^2 x \, dx$
$\int \cos^2 x \, dx$
$\int \cos^4 x \, dx$
$\int \frac{1}{2}(1 + \cos 2x) \, dx$
$\frac{1}{2}\int 1 \, dx$
$\frac{1}{2}\int \cos 2x \, dx$
$\frac{1}{2}x$
$\frac{1}{4}\sin 2x$

AND logic

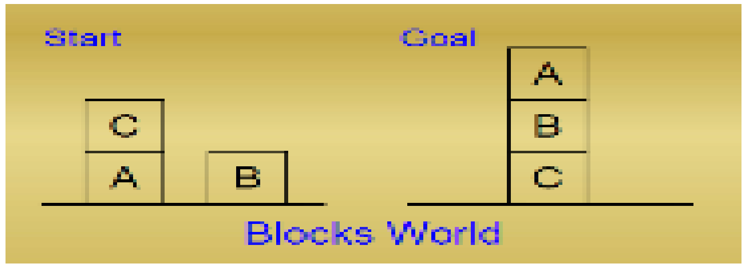
Is the problem Decomposable?

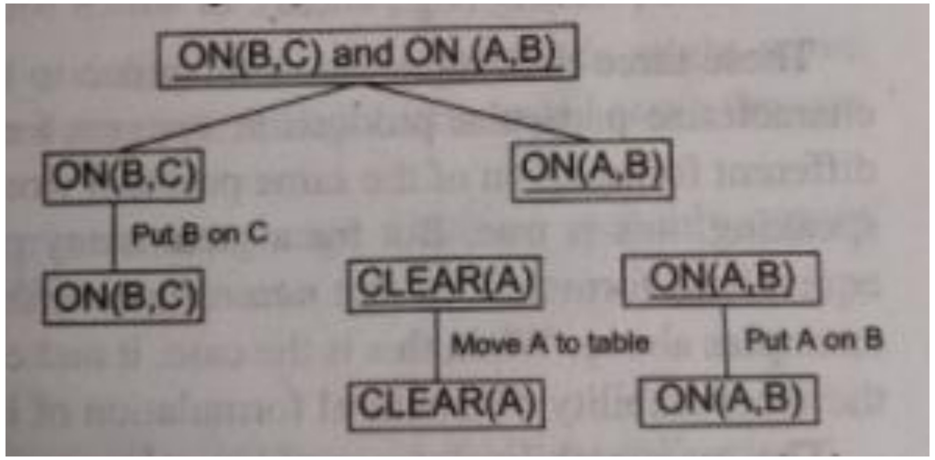
- Non - Decomposable problem:



Is the problem Decomposable?

- Non - Decomposable problem:





Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem's universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Can solution steps to be ignored or undone?

- Consider following 3 problems

1: proving a theorem or lemma

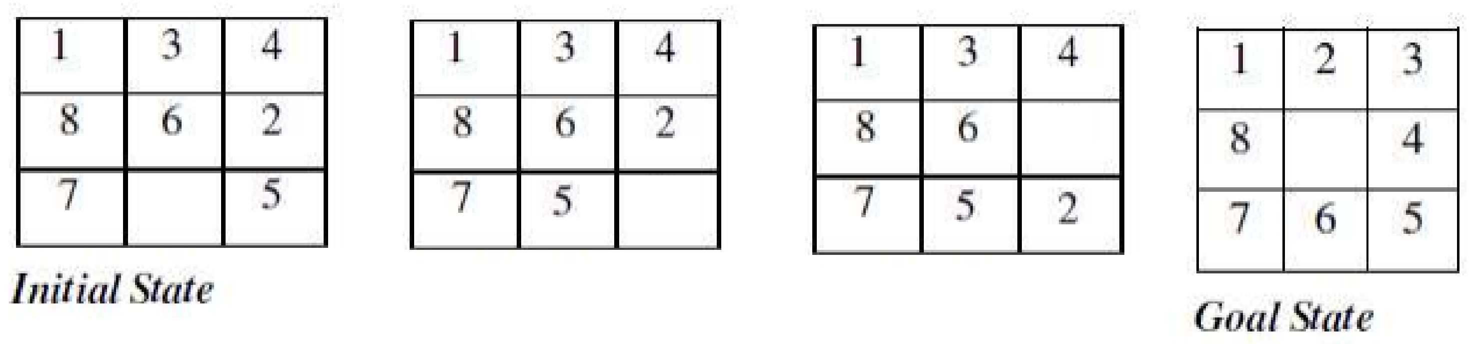
  – some steps can be ignored

  – logically derives to a solution

Can solution steps to be ignored or undone?

- Consider following 3 problems

2: 8 puzzle problem



- Recoverable problem

Can solution steps to be ignored or undone?

- Consider following 3 problems

3: chess problem

- Irrecoverable problem

Can solution steps to be ignored or undone?

- Proving a theorem or lemma
  - Ignorable
- 8 puzzle problem
  - Recoverable
- Chess game
  - Irrecoverable

  Recovery of the problem plays an important role in determining the complexity of the control structure

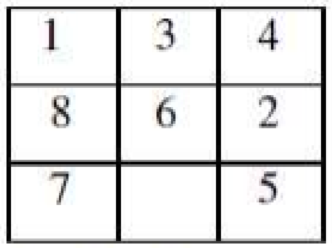
Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

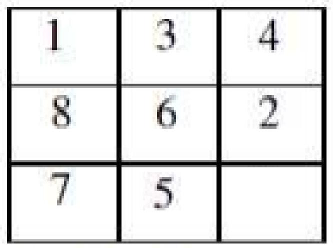
Is the problem's universe predictable?

- 8 puzzle problem – next step is always predictable – normal planning - certain outcome

- Card game – next st ility for finding – uncertain problem



1	3	4
8	6	2
7		5



1	3	4
8	6	2
7	5	

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Is the good solution is absolute or relative?

- Consider this example: Given some facts

1. Deena is a man.
2. Deena is a worker in a company.
3. Deena was born in 1905.
4. All men are mortal.
5. All workers in a factory died when there was an accident in 1952.
6. No mortal lives longer than 100 years.

“Is Deena alive”

Is the good solution is absolute or relative?

1. Deena is a man.
2. Deena is a worker in a company.
3. Deena was born in 1905.
4. All men are mortal.
5. All workers in a factory died when there was an accident in 1952.
6. No mortal lives longer than 100 years.

"Is Deena alive"

Solution 1:
1. Deena is a man.
2. Deena was born in 1905.
3. All men are mortal.
4. Now it is 2020, so Siva's age is 105 years.
5. No mortal lives longer than 100 years.

Solution 2:
1. Deena is a worker in the company.
2. All workers in the company died in 1952.

Any path problem – Relative solutions

Is the good solution is absolute or relative?

Consideration 2 : Travelling Salesman problem

Travelling salesman problem: Goal : shortest path from source city to destination city – visiting all cities one by one

Solution : Shortest path – to visit all cities

Best path problem – Absolute solutions

Is the good solution is absolute or relative?

Solution:

Any path problem – Relative solution

Best path problem – Absolute Solution

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Is the solution a state or a path?

Consideration 1: Inference from the statement

“The bank president ate a dish of pasta salad with the fork”

Inference: pasta salad was a dish , pasta salad was eaten, pasta salad consists of pasta….

Solution: state

Is the solution a state or a path?

Consideration 2 : Path problem

Water jug problem: Goal : (2,0)

Solution : path taken to reach goal state from initial state

Solution: path taken from (0,0) – (2,0)

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem's universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

What is the role of knowledge?

- **Chess playing**

  – rules for determining legal moves + some simple control mechanisms

- **News paper story**

  – scan all daily newspapers + how many supports Modi jee + how many supports Soina jee for upcoming election

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem's universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Does the task require interaction with a person?

- **Solitary**:
  - Computer is given with a problem description
  - no intermediate communication
  - no demand for an explanation
- **Conversational**:
  - Intermediate conversations between person to computer
  - User need to provide additional information
  - Robotics

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

- https://youtu.be/iaAbUhqXmNo

Problem Analysis and Representation

Solution:

- Two water jugs are there with 5 litre and 2 litre capacity and there is no measurements available
- The allowed operations are filling the jug and pouring out water from the jug
- initial state: (5,2)
- goal state: (1,0)
- **Sequence of operations:**

    i) empty big(remove water from big jug)

    ii) empty small(remove water from small jug)

    iii) big is empty(pour water from small jug to big jug)

    iv) small is empty(pour water from big jug to small jug)

    actions of sequence: 2,4,2,4,2

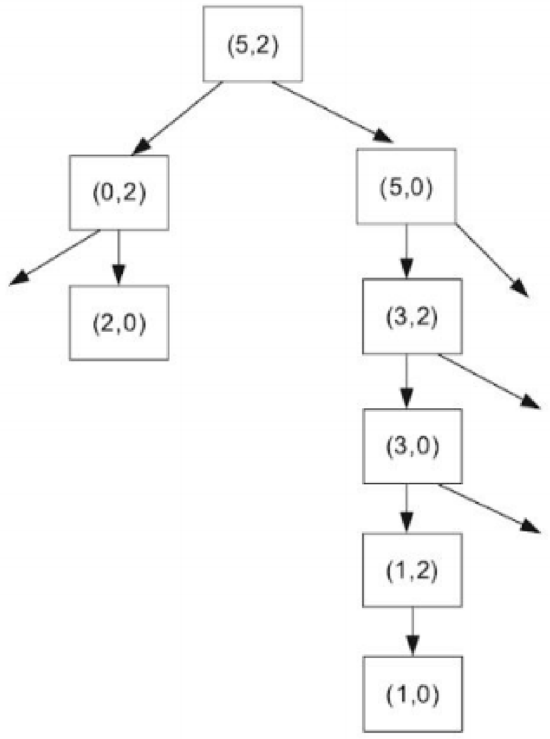


Figure 2.7 Water jug problem.

Problem Analysis and Representation

- Problem definition must satisfy these criteria:

1. **Compactness:** Solution space should be clearly defined
2. **Utility:** Compatible with existing systems
3. **Soundness:** Should not raise false alarm
4. **Completeness:** Should have all the past details
5. **Generality:** Able to handle all similar events
6. **Transparency:** Reasoning should be visible to the user

Performance Measuring

- Various factors needs to be considered in problem solving
- Three outcomes of problem solver
    - Finding a solution
    - Terminating with failure after search space is exhausted
    - Terminating after certain number of iterations
- Reward: if the problem is solved but the amount of time and resource used needs to be considered

Performance Gain

- Performance gain,
  - Problem: Well defined problem or poorly defined
  - Time: Time taken to arrive at the solution
  - Resource: Storage cost, hardware cost, etc.…
  - Result: Success or Failure

Problem Space and Search

- Problem is represented as state space
- Search is a general algorithm for finding path in state space
- The identified path will either lead to solution or dead end
- Search algorithm makes use of control strategy like forward or backward search
  - forward search(data directed)
    - Starts search from initial state towards goal state.
    - Ex: locating a city from current location
- backward search(goal directed)
  - Search stars from goal state towards a Solvable initial state
  - Ex: start from target city

Problem Space and Search

- Strategies to explore the states
  - Informed search – No guarantee for solution but high probability of getting solution
    - -heuristic approach is used to control the flow of solution path
    - -heuristic approach is a technique based on common sense, rule of thumb, educated guesses or intuitive judgment
  - Uninformed search – generates all possible states in the state space and checks for the goal state.
    - - time consuming due to large state space
    - - used where error in the algorithm has severe consequences

- Parameters for search evaluation
  - i) completeness: Guaranteed to find a solution within finite time
  - ii) space and time complexity: memory required and time factor needed
  - iii) optimality and admissibility: correctness of the solution

Informed Search

- Does not guarantee a solution
- But it ensures high probability of arriving at solution
- Heuristic is a problem specific knowledge or guidance used to constrain the search and lead to the goal
- Heuristic is based on common sense or rule of thumb, educated guesses or intuitive judgement
- It helps us to choose the right path when multiple path exist for a problem

Uninformed Search

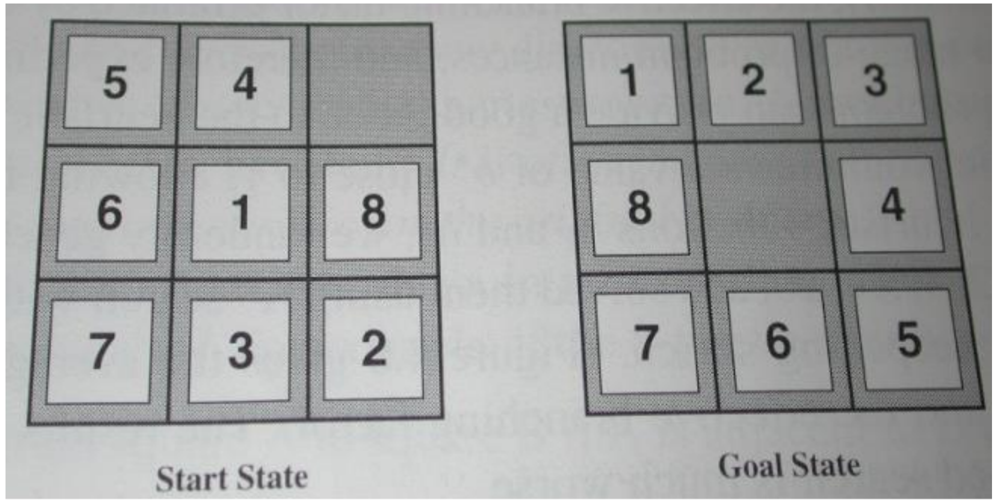
- Uninformed search is also referred as blind search
- Generates all possible states in state space and checks for goal state
- It will always find a solution if it exists
- But the method is time consuming since search space is huge
- It is used to benchmark results of other algorithms

Problems in design of search programs

- State representation and identifying relationships between states
- Proper selection of forward and backward movement to determine optimal path
- Rule selection

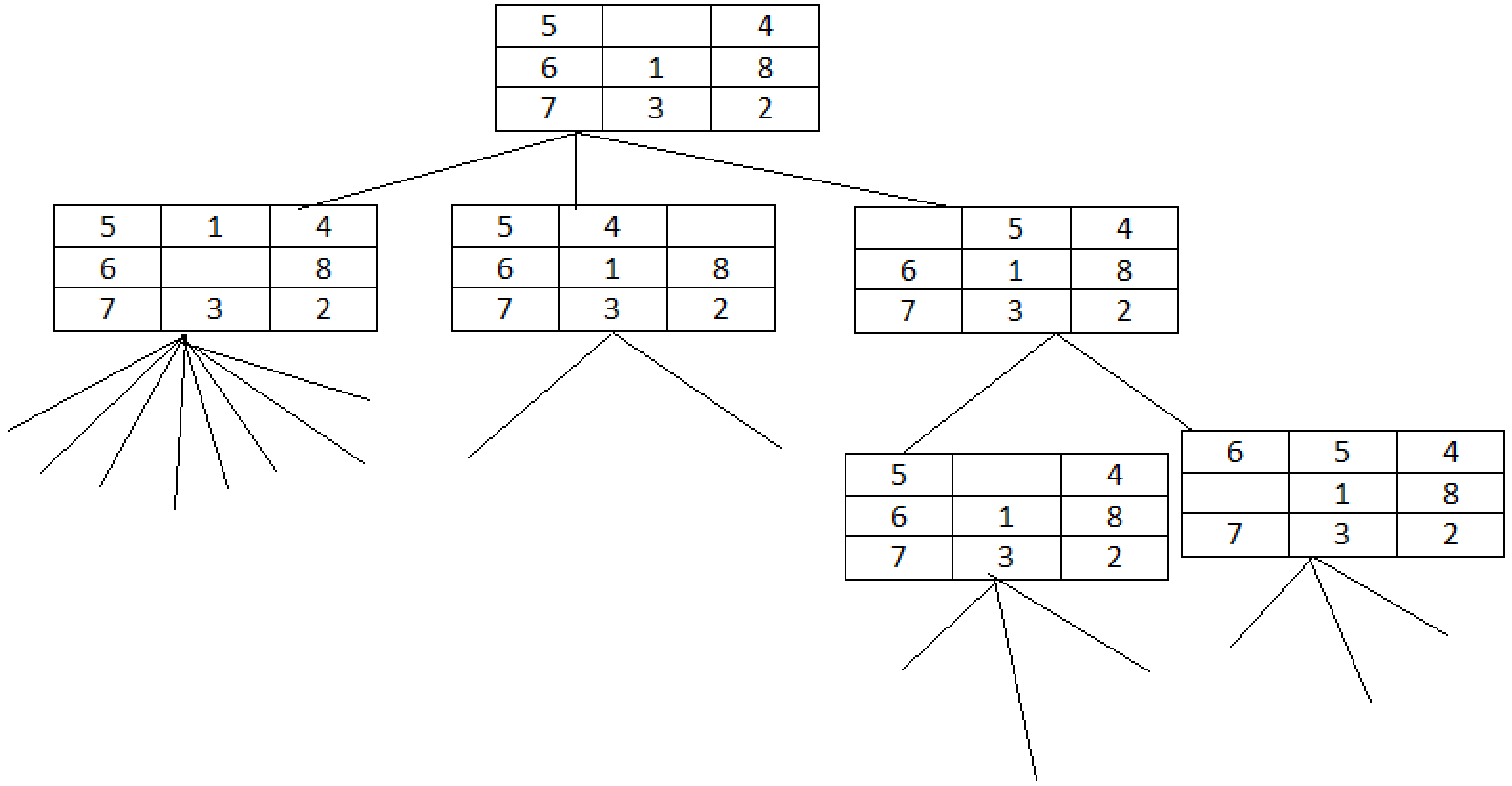
Toy problems

1. **8 puzzle problem**
    - 3x3 board with eight numbered tiles and a blank space.
    - A tile adjacent to the blank space can slide into the space.
    - objective-to reach the configuration shown on the right of the figure.



Toy problems

1. 8 puzzle problem – search space



Toy problems

1. 8 puzzle problem

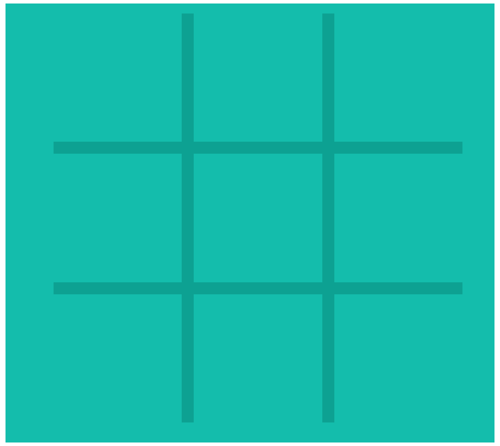
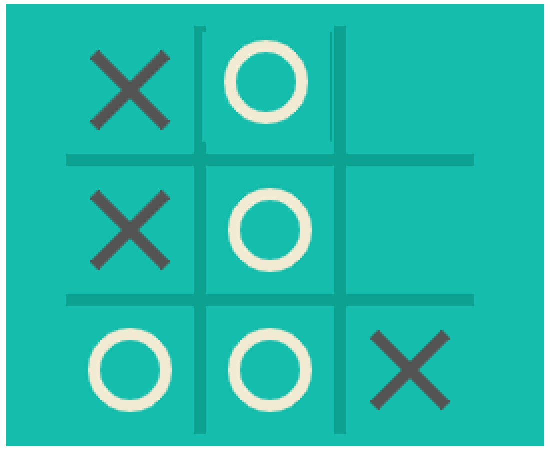
Problem formulation:

- **States:** a state description specifies the location of each of the eight tiles in one of the nine squares. For efficiency, it is useful to include the location of the blank.
- **Initial state:** Numbers are not arranged in clockwise order
- **Operators:** blank moves left, right, up, or down.
- **Goal state:** state matches the goal configuration shown in previous Figure
- **Path cost:** each step costs 1, so the path cost is just the length of the path.

Toy problems

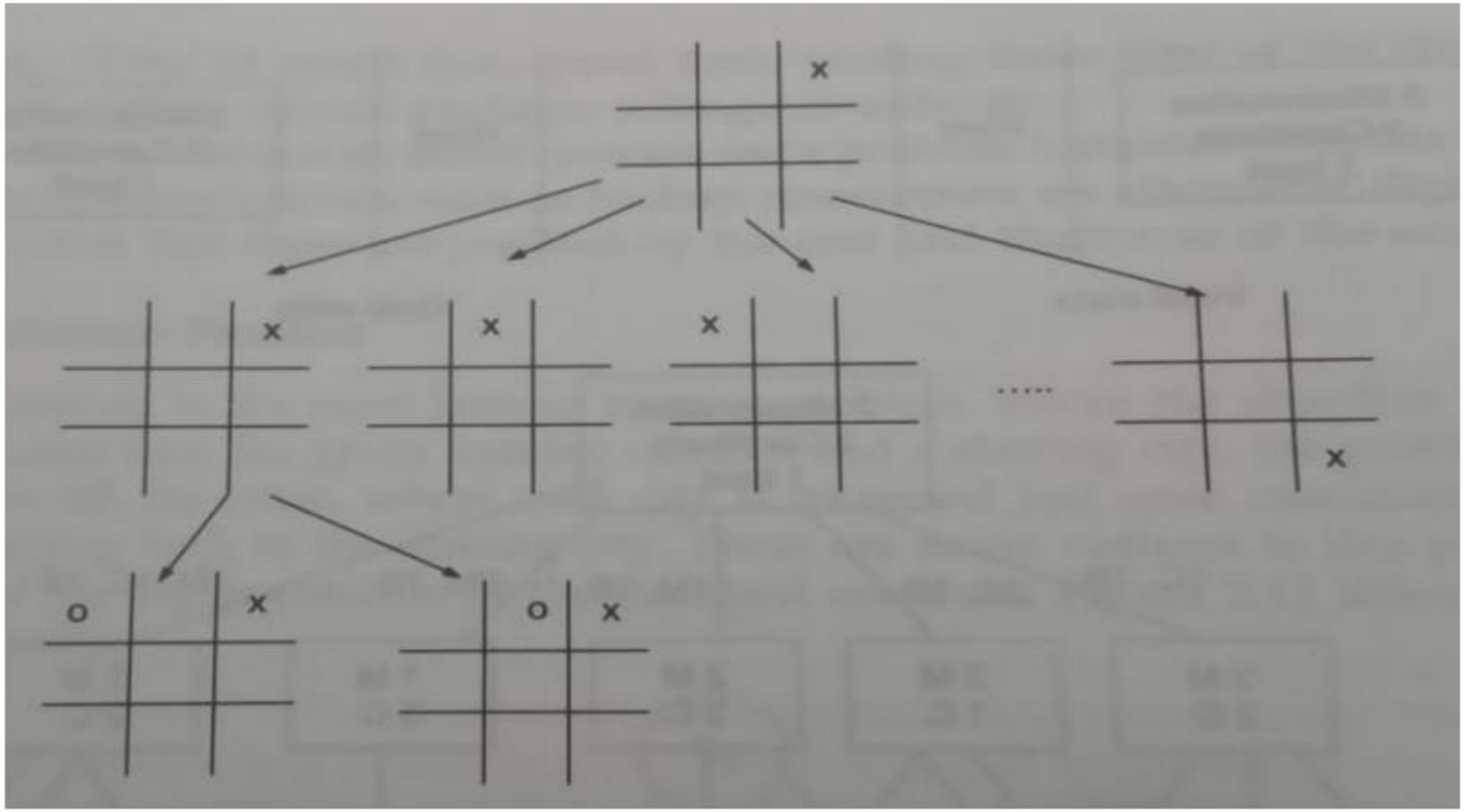
2. Tic-tac-toe problem

- Each player marks a 3*3 grid by 'x' and 'o' in turn.
- The player who puts respective mark in a horizontal, vertical or diagonal line wins the game
- If both players fail to reach above criteria and all boxes in the grid are filled, then the game is draw

Toy problems

2. Tic-tac-toe problem



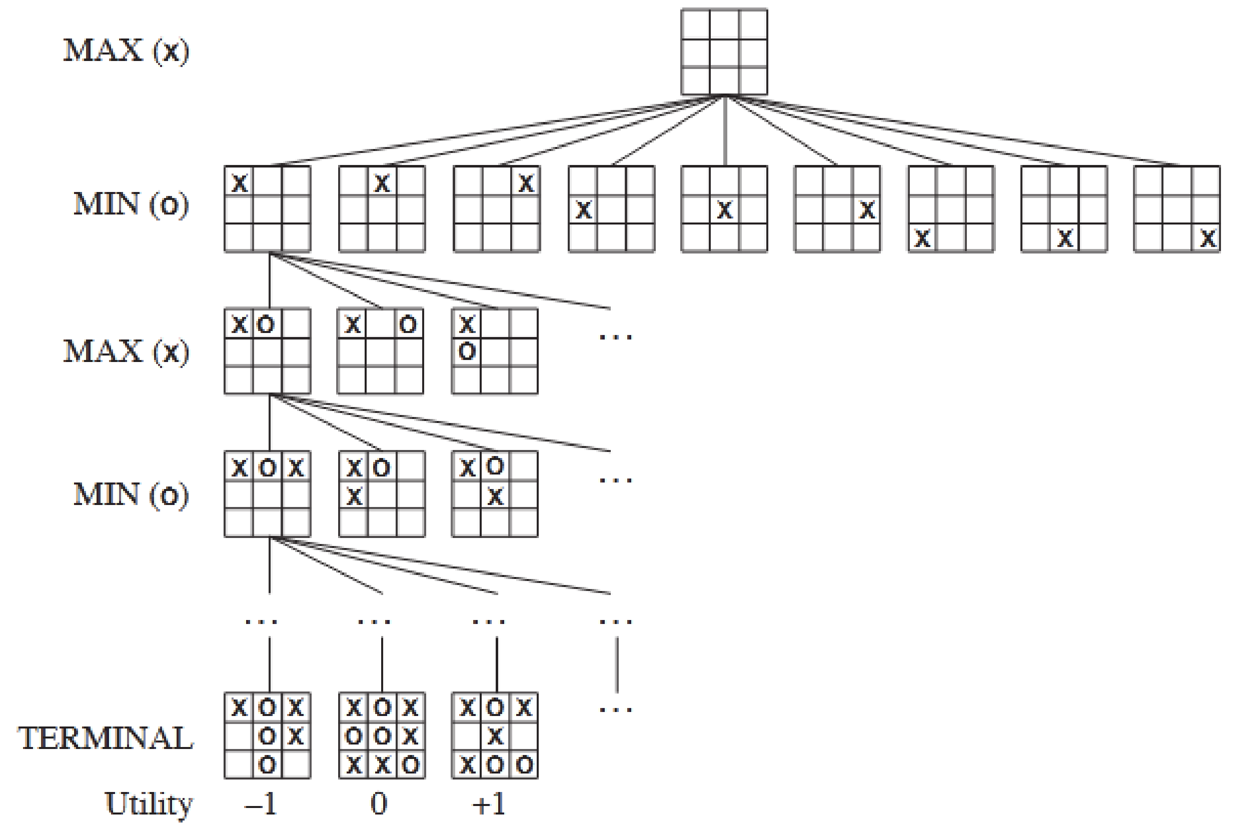
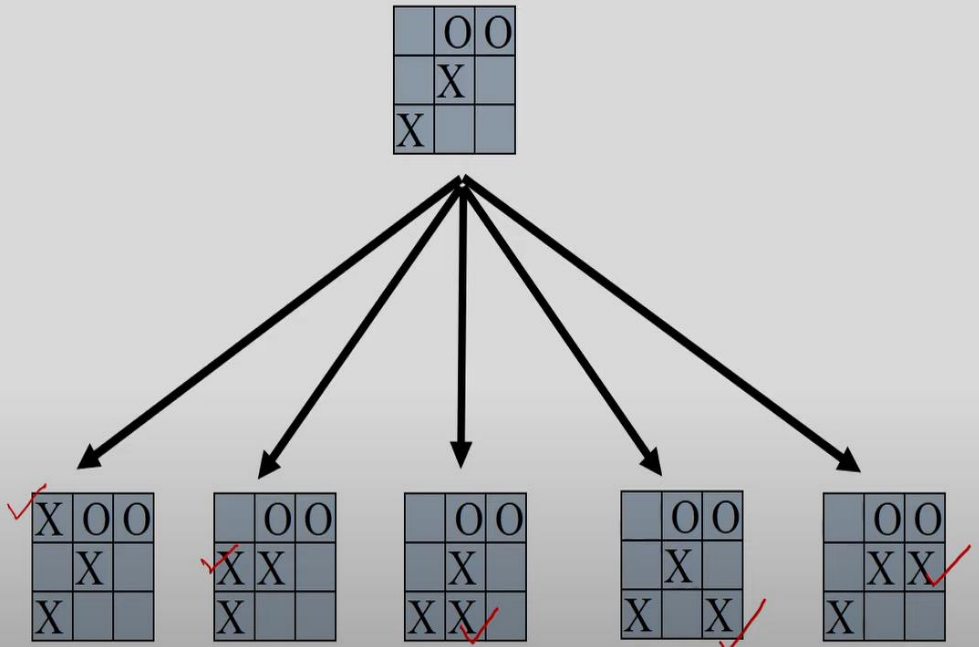


Figure 5.1 A (partial) game tree for the game of tic-tac-toe. The top node is the initial state, and MAX moves first, placing an X in an empty square. We show part of the tree, giving alternating moves by MIN (O) and MAX (X), until we eventually reach terminal states, which can be assigned utilities according to the rules of the game.

- Use function to evaluate the goodness of a state; like the heuristics functions for informed search.
  - This function is called the evaluation function.
  - Evaluation function is applied to the intermediate node.

Tic-Tac-Toe

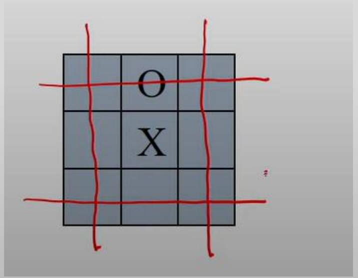


Evaluation Function

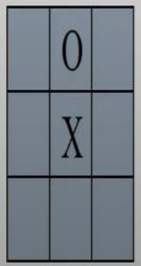
e(p) = number of directions open for Max – number of directions open for Min

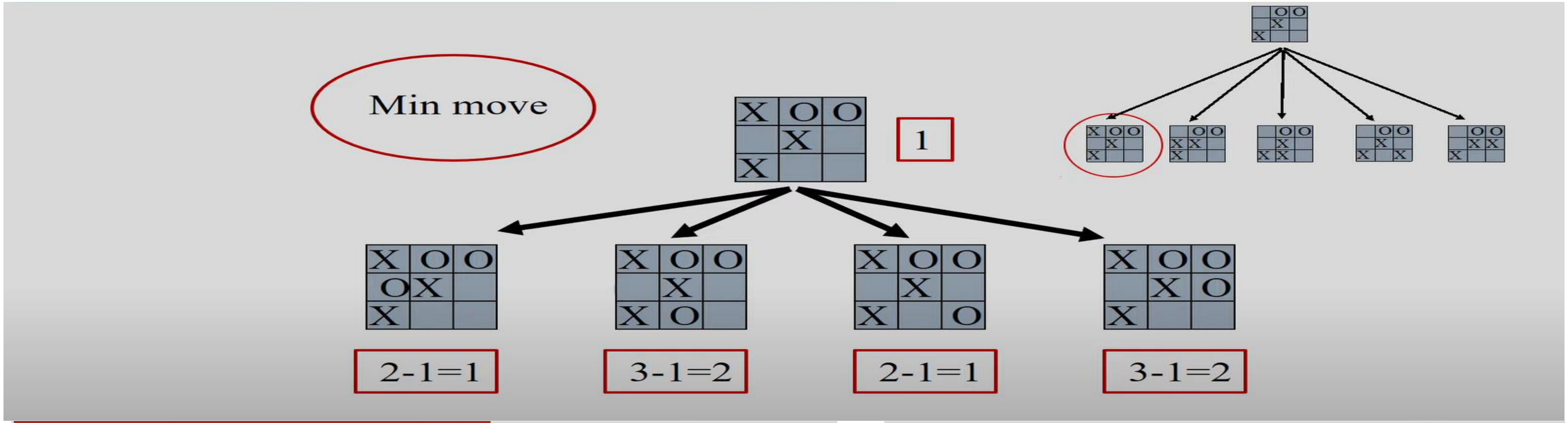
e(p) = + inf if win for Max

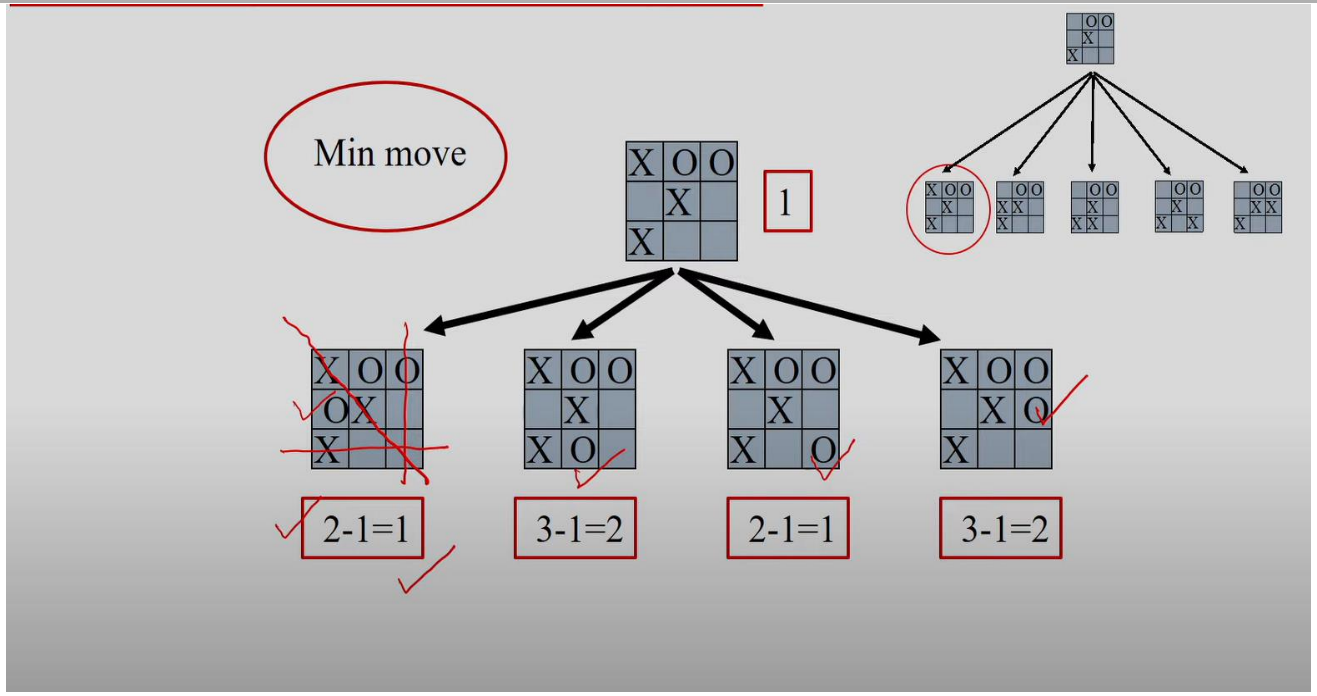
e(p) = - inf if win for Min

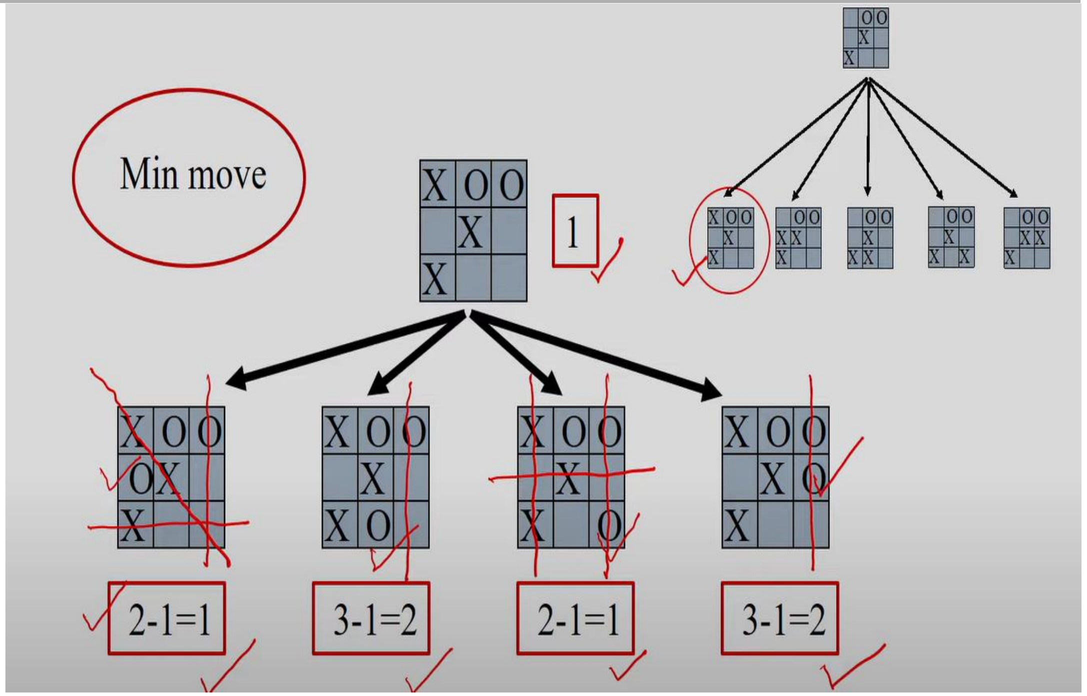


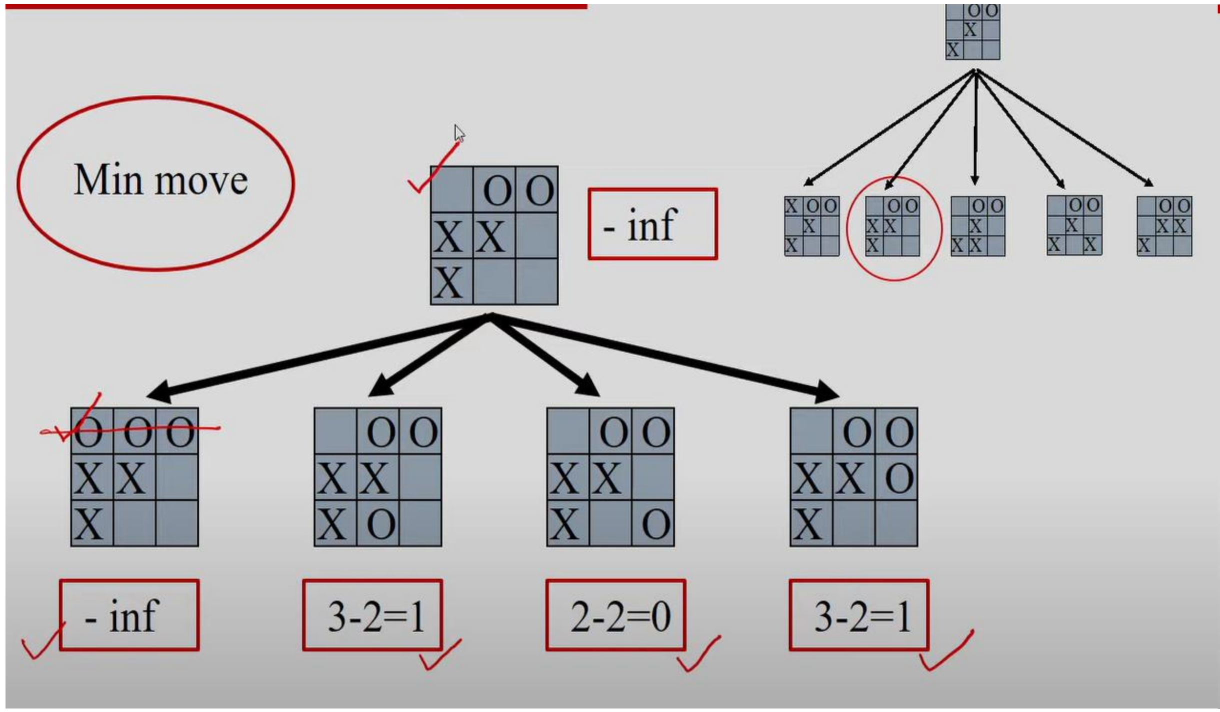
e(p) = 6 – 4 = 2

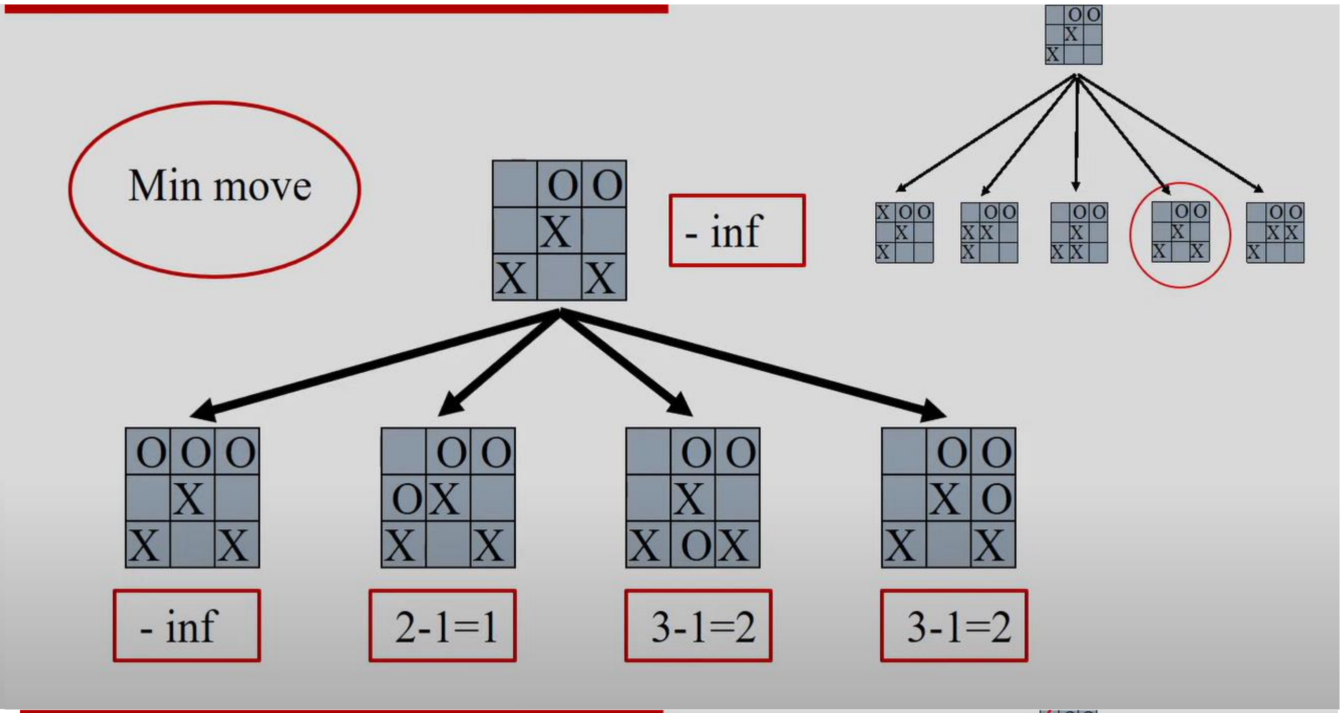


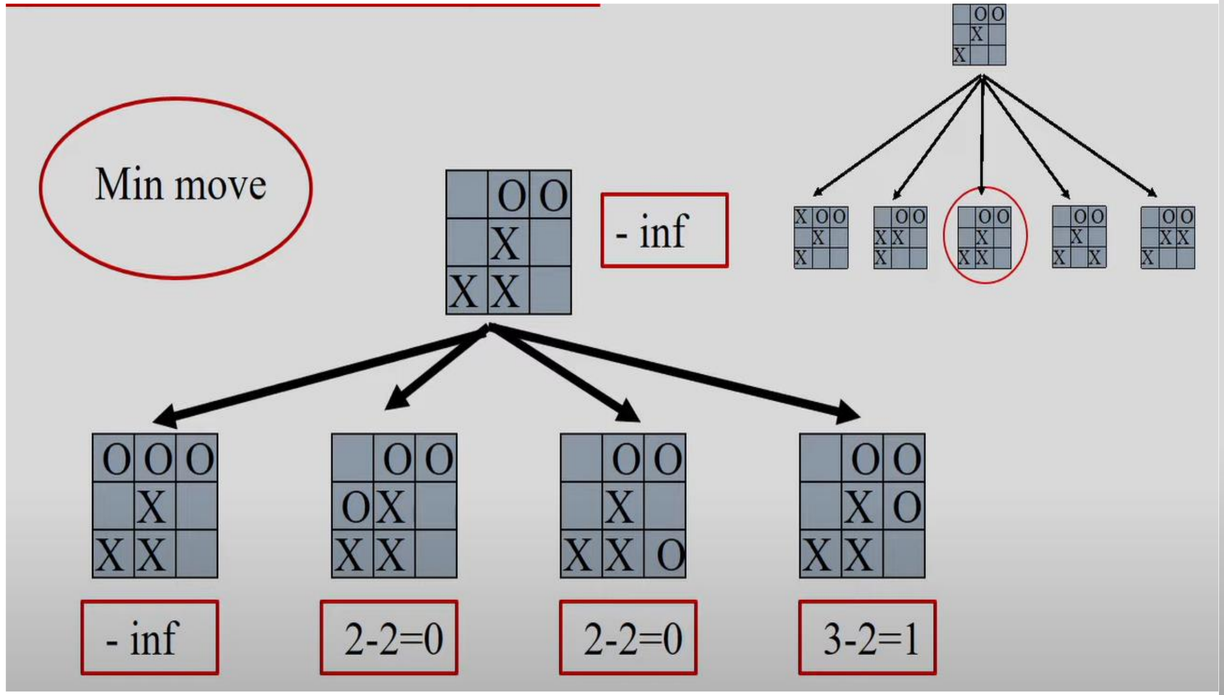

Min move
1
2-1=1
3-1=2
2-1=1
3-1=2

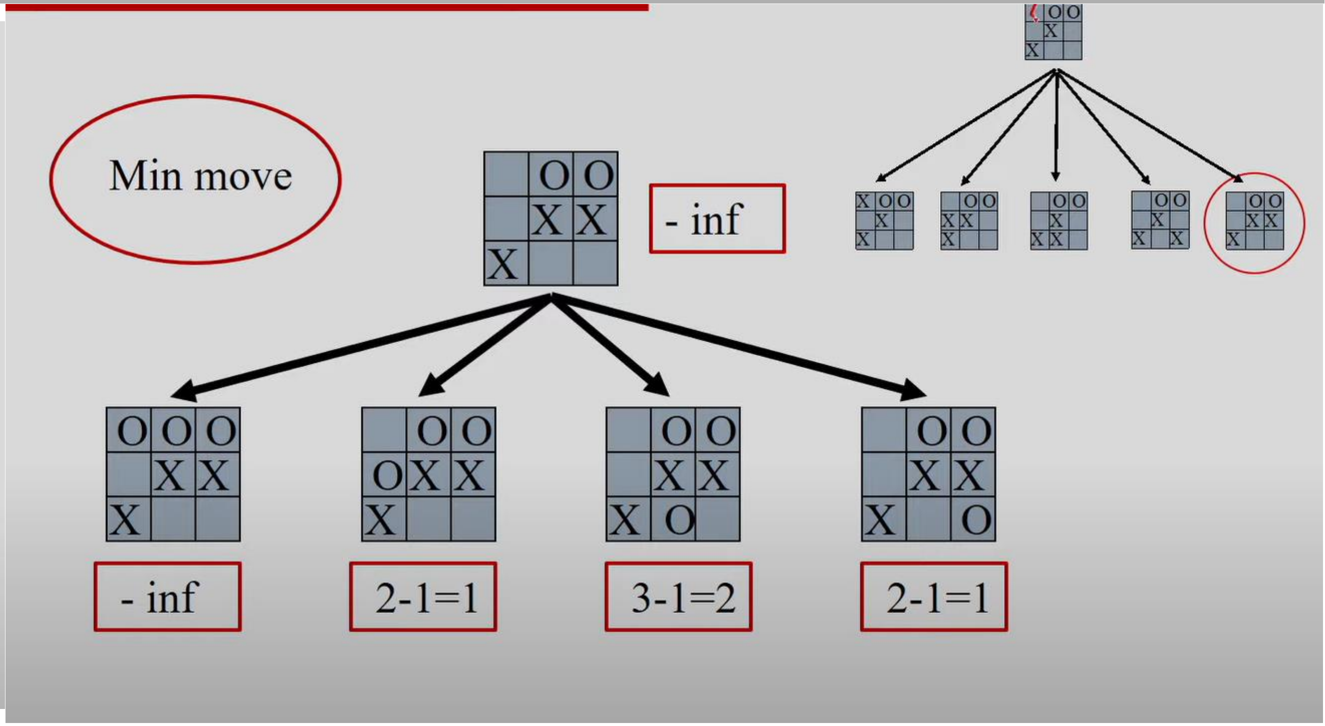

Min move
1
2-1=1
3-1=2
2-1=1
3-1=2


Min move
1
2-1=1
3-1=2
2-1=1
3-1=2


Min move
- inf
- inf
3-2=1
2-2=0
3-2=1


Min move
- inf
- inf
2-1=1
3-1=2
3-1=2


Min move
- inf
- inf
2-2=0
2-2=0
3-2=1


Min move
- inf
- inf
2-1=1
3-1=2
2-1=1

Toy problems

2. Tic-tac-toe problem

Formulating Tic-tac-toe problem

- **Initial state** – state in previous figure
- **States** – Next figure with 'x' and 'o' positions constitutes the states in space
- **Operators** – Adding 'x' or 'o' in cells one by one
- **Goal** – To reach final/winning position
- **Path cost** – Each step costs 1 so that path cost is length of path

Toy problems

3. Missionaries and Cannibals

- Three missionaries and cannibals
- Need to move all the six people from one bank of river to the other
- The boat has a capacity of one of two people

Toy problems

3. Missionaries and Cannibals

Formulating the problem in state space search

- States – sequence of 3 numbers representing number of missionaries, cannibals and boat.
- Goal state - missionaries and cannibals have reached other side of river (3,3,1)
- Initial state - (3,3,0)
- Operator - Putting missionary and cannibal in boat such that cannibal does not outnumber missionary and one/two people in boat
- Path cost – number of crossings

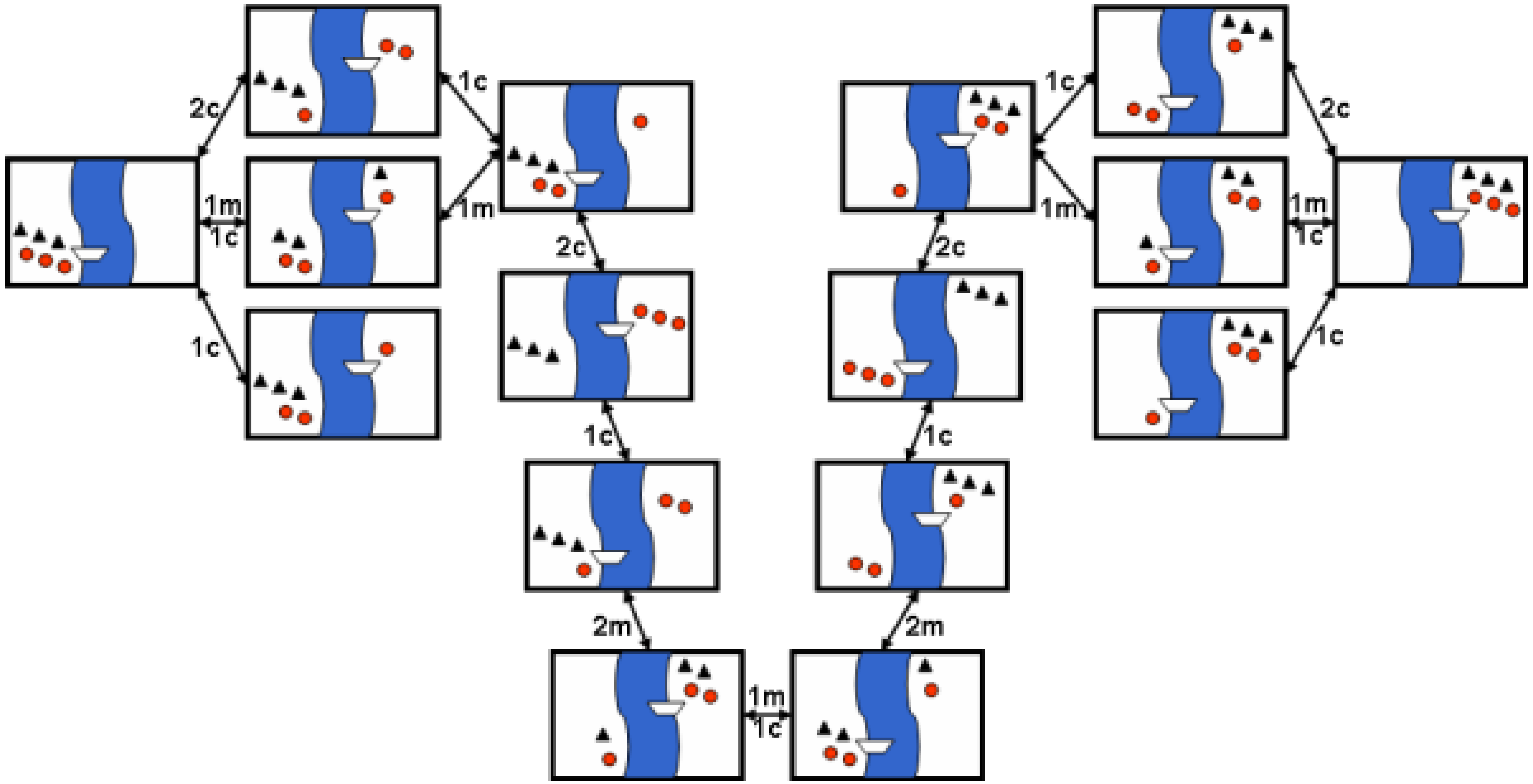
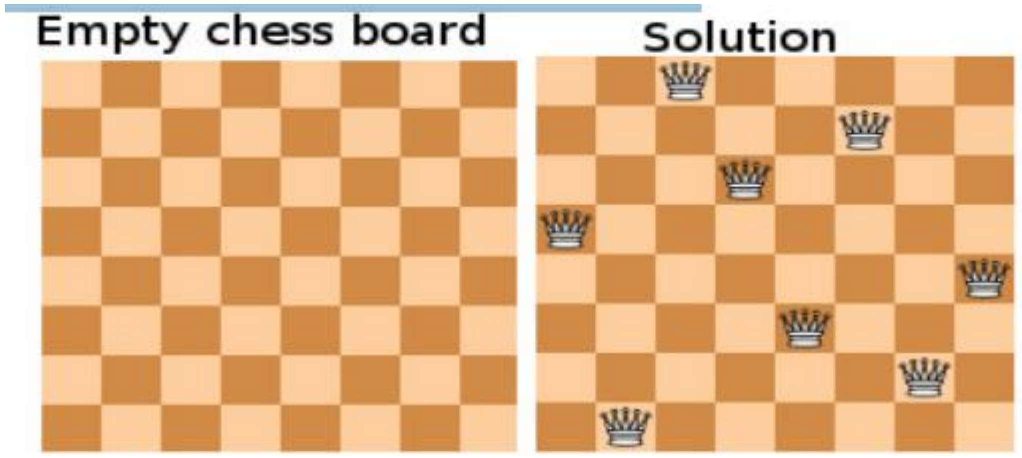


Figure 1: Search-space for the Missionaries and Cannibals problem

Toy problems

4. The 8-queens problem

- Place eight queens on a chessboard such that no queen attacks any other.
- There are two main kinds of formulation
- The incremental formulation involves placing queens one by one
- The complete-state formulation starts with all 8 queens on the board and moves them around.
- Goal test: 8 queens on board, none attacked



Toy problems

4. The 8-queens problem

Formulating the problem in state space search

Consider the following for incremental formulation:

- States: Any arrangement of 0 to 8 queens on board.
- Initial state: No queens in the board
- Goal state: Queens in each column without targeting other queens
- Operators: add a queen to any square.
- Path cost: Number of moves

Toy problems

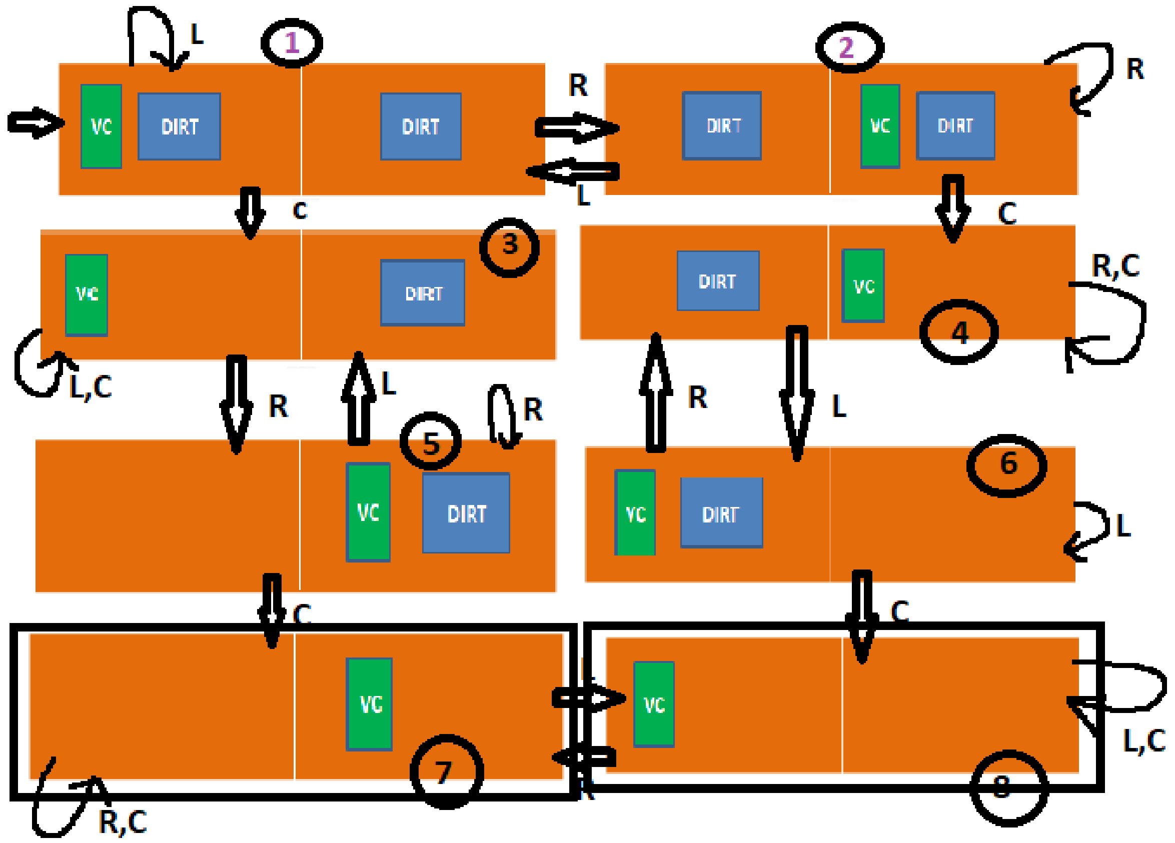
5. Vacuum cleaner problem

Assume that the agent knows its location and the locations of all the pieces of dirt, and the suction is still in good working order.

- States: It is based on Vacuum cleaner location and dirt location
- Initial state: Any state can be assumed as initial state
- Operators: move left, move right, suck.
- Goal state: no dirt left in any square.
- Path cost: each action costs 1.

Toy problems

5. Vacuum cleaner problem



Real-world problems

- **Route finding:**
- Defined in terms of locations and transitions along links between them
- Applications: routing in computer networks, automated travel advisory systems, airline travel planning systems

Real-world problems

- **Touring and traveling salesperson problems:**
- “Visit every city on the map at least once and end in Bucharest”
- Needs information about the visited cities
- Goal: Find the shortest tour that visits all cities
    - NP-hard, but a lot of effort has been spent on improving the capabilities of TSP algorithms

Real-world problems

- **VLSI layout:**
  - Place cells on a chip so they don't overlap and there is room for connecting wires to be placed between the cells
- **Robot navigation:**
  - Generalization of the route finding problem
  - No discrete set of routes
  - Robot can move in a continuous space
  - Infinite set of possible actions and states

Real-world problems

- **Assembly sequencing:**
  - Automatic assembly of complex objects
  - The problem is to find an order in which to assemble the parts of some object