

MODULE 1

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

What is AI?

Artificial Intelligence (AI) is a branch of Science which deals with helping machines find solutions to complex problems in a more human-like fashion. This generally involves borrowing characteristics from human intelligence, and applying them as algorithms in a computer friendly way.

A more or less flexible or efficient approach can be taken depending on the requirements established, which influences how artificial the intelligent behavior appears. Artificial Intelligence is a way of **making a computer, a computer-controlled robot, or a software think intelligently**, in the similar manner the intelligent humans think. AI is accomplished by studying how human brain thinks and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

Goals of AI

To Create Expert Systems – The systems which exhibit intelligent behavior, learn, demonstrate, explain, and advise its users.

To Implement Human Intelligence in Machines – Creating systems that understand, think, learn, and behave like humans.

APPLICATIONS OF ARTIFICIAL INTELLIGENCE

1. Robotic vehicles

A self-driving car, also known as an autonomous vehicle (AV), driverless car, or robocar is a vehicle that is capable of sensing its environment and moving safely with little or no human input. Self-driving cars combine a variety of sensors to perceive their surroundings. Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles and relevant signage.

2. Speech recognition

Speech recognition is concerned with the recognition and translation of spoken language into text by computers. It is also known as automatic speech recognition, computer speech recognition or speech to text. Some speech recognition systems require “training” where an individual speaker reads text or isolated vocabulary into the system. The system analyses the person’s specific voice and uses it to fine-tune the recognition of that person’s speech, resulting in increased accuracy.

3. Autonomous planning and scheduling

NASA’s Remote Agent program was the first on-board autonomous planning program to control the scheduling of operations for a spacecraft. The Remote Agent generated plans from high-level goals specified from the ground and monitored the execution of those plans - detecting, diagnosing, and recovering from problems as they occurred. Successor program MAPGEN planned the daily operations for NASA’s Mars Exploration Rovers, and MEXAR2 did mission planning for the European Space Agency’s Mars Express mission in 2008.

4. Game playing

IBM’s DEEP BLUE became the first computer program to defeat the world champion in a chess match when it defeated Garry Kasparov in an exhibition match in 1997.

5. Spam fighting

Each day, learning algorithms classify over a billion messages as spam, saving the recipient from having to waste time deleting them.

6. Logistics planning

During the Persian Gulf crisis of 1991, U.S. forces deployed a Dynamic Analysis and Replanning Tool (DART) to do automated logistics planning and scheduling for transportation. This involved up to 50,000 vehicles, cargo, and people at a time, and had to account for starting points, destinations, routes, and conflict

resolution among all parameters. The artificial intelligence planning techniques generated in hours a plan that would have taken weeks with older methods.

7. Robotics

The iRobot Corporation has sold over two million Roomba robotic vacuum cleaners for home use. The company also deploys the more rugged PackBot to Iraq and Afghanistan, where it is used to handle hazardous materials, clear explosives, and identify the location of snipers.

8. Machine translation

There are computer programs that can automatically translate from, say, Arabic to English. The program uses a statistical model built from examples of Arabic-to-English translations and from examples of English text totalling two trillion words.

Recent applications

The following are some of the recent applications of artificial intelligence.

1. Agriculture

In agriculture new artificial intelligence advancements show improvements in gaining yield and to increase the research and development of growing crops. New artificial intelligence methods now predict the time it takes for a crop like a tomato to be ripe and ready for picking thus increasing efficiency of farming.

2. Cybersecurity

The more advanced of these solutions use artificial intelligence and natural language processing (NLP) methods to automatically sort the data in networks into high risk and low-risk information. This enables security teams to focus on the attacks that have the potential to do real harm to the organization, and not become victims of attacks such as denial-of-service (DoS), malware and others.

3. Finance

Banks use artificial intelligence systems today to organize operations, maintain bookkeeping, invest in stocks, and manage properties. Artificial intelligence can react to changes overnight or when business is not taking place.

4. Healthcare

The primary aim of health-related artificial intelligence applications is to analyse relationships between prevention or treatment techniques and patient outcomes. Artificial intelligence programs are applied to practices such as diagnosis processes, treatment protocol development, drug development, personalized medicine, and patient monitoring and care. Artificial intelligence algorithms can also be used to analyse large amounts of data through electronic health records for disease prevention and diagnosis.

What is Intelligence?

The ability of a system to calculate, reason, perceive relationships and analogies, learn from experience, store and retrieve information from memory, solve problems, comprehend complex ideas, use natural language fluently, classify, generalize, and adapt new situations.

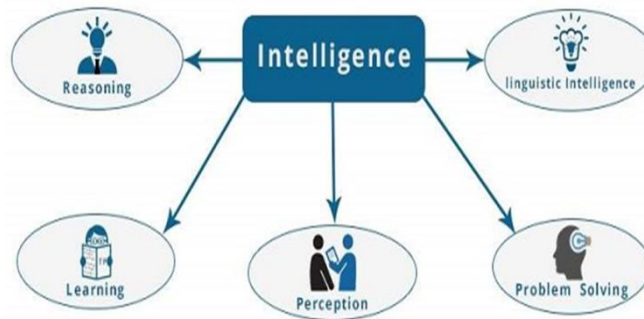
Types of Intelligence

- Linguistic intelligence
- Musical intelligence
- Logical-mathematical intelligence
- Spatial intelligence
- Intra-personal intelligence
- Interpersonal intelligence

What is Intelligence Composed of?

It is composed of –

- Reasoning
- Learning
- Problem Solving
- Perception
- Linguistic Intelligence



Reasoning

It is the set of processes that enables us to provide basis for judgement, making decisions, and prediction.

There are broadly two types –

- Inductive Reasoning
- Deductive Reasoning

Inductive Reasoning	Deductive Reasoning
It conducts specific observations to makes broad general statements.	It starts with a general statement and examines the possibilities to reach a specific, logical conclusion.
Even if all of the premises are true in a statement, inductive reasoning allows for the conclusion to be false.	If something is true of a class of things in general, it is also true for all members of that class.
Example – "Nita is a teacher. All teachers are studious. Therefore, Nita is studious."	Example – "All women of age above 60 years are grandmothers. Shalini is 65 years. Therefore, Shalini is a grandmother."

Learning –

It is the activity of gaining knowledge or skill by studying, practicing, being taught, or experiencing something. Learning enhances the awareness of the subjects of the study. The ability of learning is possessed by humans, some animals, and AI-enabled systems.

Auditory Learning – It is learning by listening and hearing.

For example, students listening to recorded audio lectures.

Episodic Learning – To learn by remembering sequences of events that one has witnessed or experienced. This is linear and orderly.

Motor Learning – It is learning by precise movement of muscles.

For example, picking objects, Writing, etc.

Observational Learning – To learn by watching and imitating others.

For example, child tries to learn by mimicking her parent.

Perceptual Learning – It is learning to recognize stimuli that one has seen before.

For example, identifying and classifying objects and situations.

Stimulus-Response Learning – It is learning to perform a particular behavior when a certain stimulus is present.

For example, a dog raises its ear on hearing doorbell.

Relational Learning – It involves learning to differentiate among various stimuli on the basis of relational properties, rather than absolute properties.

For Example, Adding 'little less' salt at the time of cooking potatoes that came up salty last time, when cooked with adding say a tablespoon of salt.

Spatial Learning – It is learning through visual stimuli such as images, colors, maps, etc.

For Example, A person can create roadmap in mind before actually following the road.

Problem Solving

It is the process in which one perceives and tries to arrive at a desired solution from a present situation by taking some path, which is blocked by known or unknown hurdles.

Problem solving also includes decision making, which is the process of selecting the best suitable alternative out of multiple alternatives to reach the desired goal are available.

Perception

It is the process of acquiring, interpreting, selecting, and organizing sensory information. Perception assumes sensing. In humans, perception is aided by sensory organs. In the domain of AI, perception mechanism puts the data acquired by the sensors together in a meaningful manner.

Linguistic Intelligence

It is one's ability to use, comprehend, speak, and write the verbal and written language. It is important in interpersonal communication.

AI PROBLEMS

AI problems are classified as

- Area of Mathematics

To prove Mathematical theorems using games

Geometry

Game playing and theorem proving share the property that the people who do them well are considered to be displaying intelligence

- Commonsense reasoning

Includes

Reasoning physical objects and their relationships to each other

An object can be in only one place at a time

Reasoning about actions and their consequences

If you let go of something, it will fall to the floor and may be break

- Perception

Perception around our world is crucial our survival. Perceptual tasks are difficult because they involve analog signals rather than digital signals.

The analog signals are typically very noisy and usually a large number of things must be perceived at once.

- Natural Language Understanding

The ability to use language to communicate a wide variety of ideas is perhaps the most important thing. The problem of understanding spoken

language is a perceptual problem and is hard to solve for some reasons.
We can simplify the problem by restricting it to written language.

PROBLEMS, PROBLEM SPACES & SEARCH

To build a system to solve a particular problem, we need to do four things

1. Define the problem exactly: this definition must include precise specification of what the initial situation will be as well as what final situations constitute acceptable solutions to the problem.
2. Analyze the problem: A very few important features can have an immense impact on the appropriateness of various possible techniques for solving the problem.
3. Isolate and represent the task knowledge that is necessary to solve the problem.
4. Choose the best problem- solving technique and apply it to the particular problem

Problem

“It is the question which is to be solved. For solving the problem it needs to be precisely defined. The definition means, defining the start state, goal state, other valid states and transitions”.

A problem is defined formally by the following **five** components:

1. Initial state

The **initial state** that the agent starts in.

2. Actions

A description of the possible **actions** available to the agent. Given a particular state, **ACTIONS(s)** returns the set of actions that can be executed in s. We say that each of these actions is **applicable** in s.

3. Transition model

A description of what each action does. The formal name for this is the **transition model**, specified by a function **RESULT(s; a)** that returns the state

that results from doing action ***a*** in state ***s***. We also use the term **successor** to refer to any state reachable from a given state by a single action. The initial state, the actions, and the transition model all taken together implicitly define the **state space** of the problem – the set of all states reachable from the initial state by any sequence of actions. The state space forms a directed network or **graph** in which the nodes are states and the links between nodes are actions. A **path** in the state space is a sequence of states connected by a sequence of actions.

4. Goal test

The **goal test**, determines whether a given state is a goal state. Sometimes there is an explicit set of possible goal states, and the test simply checks whether the given state is one of them.

5. Path cost

A **path cost** function that assigns a numeric cost to each path. The problem-solving agent chooses a cost function that reflects its own performance measure. We assume that the cost of a path can be described as the sum of the costs of the individual actions along the path.

A **solution** to a problem is an action sequence that leads from the initial state to a goal state. The **optimal solution** is the solution having the lowest path cost among all solutions.

State space search

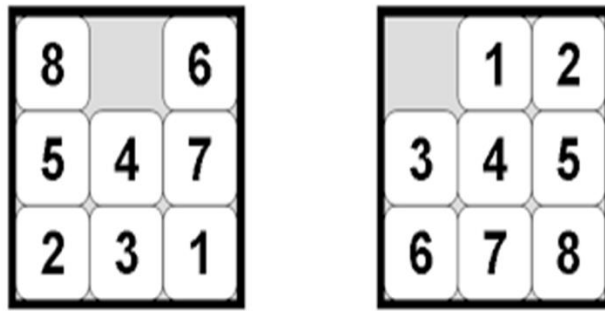
“It is complete set of states including start and goal states, where the answer of the problem is to be searched”.

A state space representation allows for the formal definition of a problem which makes the movement from initial state to the goal state quite easily. So we can say that various problems like planning, learning, theorem proving etc. are all essentially search problems only.

State space search is a process used in the field of computer science, including artificial intelligence (AI), in which successive configurations or states of an instance are considered, with the goal of finding a goal state with a desired property.

8-puzzle problem

The 8-puzzle consists of a 3 X 3 board with eight numbered tiles and a blank space. A tile adjacent to the blank space can slide into the space. The objective is to reach a specified goal state such as the one shown in Figure by sliding the tiles one at a time, given some initial or start state as in Figure.



Standard formulation

States: A state description specifies the location of each of the eight tiles and the blank in one of the nine squares. (The total number of states is $9! = 362880$.)

Initial state: Any state can be designated as the initial state.

Actions: Actions may be defined as movements of the blank space Left, Right, Up, or Down. Different subsets of these are possible depending on where the blank is.

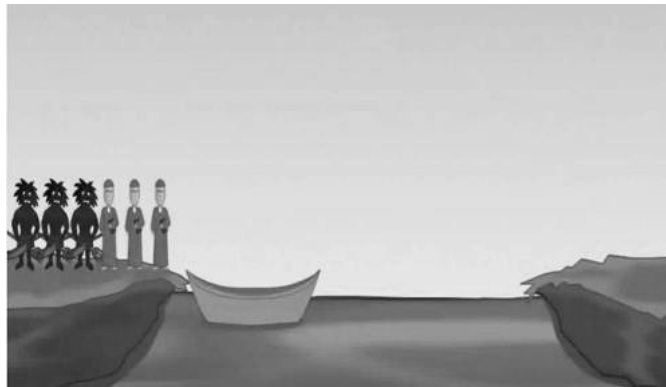
Transition model: Given a state and action, this returns the resulting state; for example, if we apply Left to the start state in Figure, the resulting state has the 5 and the blank cell switched.

Goal test: This checks whether the state matches the goal configuration shown in Figure.

Path cost: Each step costs 1, so the path cost is the number of steps in the path.

Missionaries and cannibals problem

Three missionaries and three cannibals² are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place



Standard formulation

Ignore all irrelevant parts of the problem (e.g., weather conditions, possible presence of crocodiles in the river, etc.).

States: Each state is represented by an ordered sequence of 3 numbers $(x; y; z)$ where

x : number of missionaries on initial river bank

y : number of cannibals on initial river bank

z : number of boats on initial river bank

For example, if triangles represent missionaries, circles represent cannibals and the left bank is the initial bank then the state shown in Figure is specified by the ordered triple $(0; 1; 0)$. Note that the fact that the boat is at the destination bank is indicated by $z = 0$. Also it may be noted that not all such ordered triples represent legal states. For example, the triple $(2; 3; 1)$ is not allowed because it represents a state where the number of cannibals outnumber the missionaries.

Initial state: The ordered sequence = (3; 3; 1).

Actions: Take two missionaries, two cannibals, or one of each across in the boat. We have to take care to avoid illegal states.

Transition model: Given a state and action, this returns the resulting state; for example, if we apply the action "Take 1 cannibal" to the state (0; 2; 0) the resulting state is (0; 3; 1).

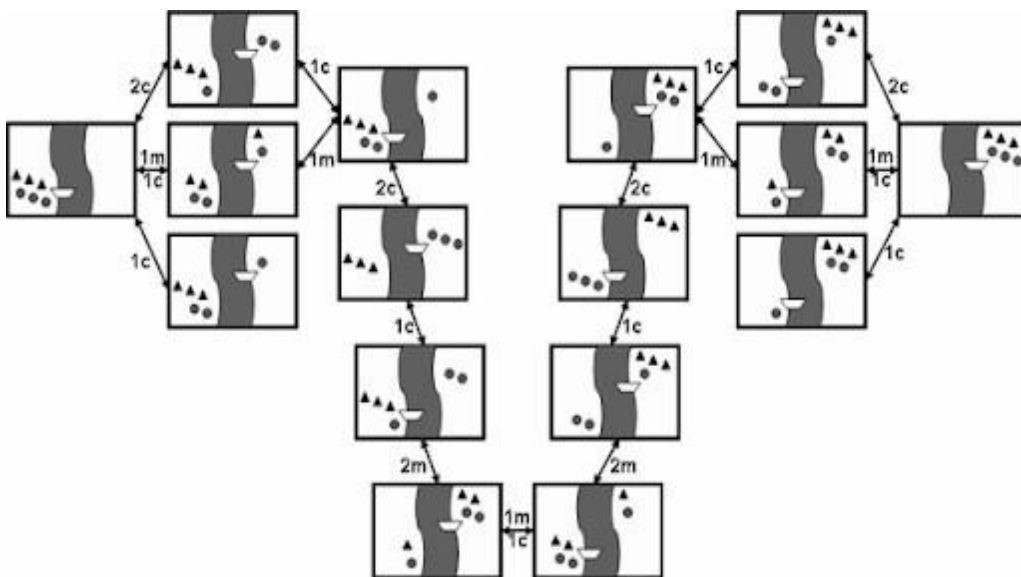
Goal test: We have reached state (0; 0; 0).

Path cost: Number of crossings.

Solution

Figure shows the possible legal states reachable from the initial state applying only legal actions. In the figure, the notation "**-2c →**" indicates the operation of taking 2 cannibals from the bank where the boat is currently located to the other bank and the notation "**-2m →**" indicates taking two missionaries. From the figure we can easily get the solution to the problem as follows:

(3; 3; 1) **-2c →** (3; 1; 0) **-1c →** (3; 2; 1) **-2c →** (3; 0; 0) **-1c →** (3; 1; 1) **-2m →** (1; 1; 0) **-1m, 1c →** (2; 2; 1) **-2m →** (0; 2; 0) **1c →** (0; 3; 1) **-2c →** (0; 1; 0) **-1c →** (0; 2; 1) **-2c →** (0; 0; 0)

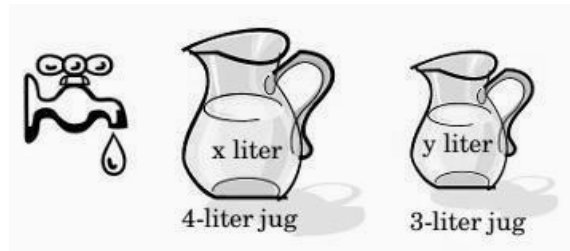


	Left		Right
1	(3; 3; 1)		(0; 0; 0)
2	(3; 1; 0)	2C ->	(0; 2; 1)
3	(3; 2; 1)	<- 1C	(0; 1; 0)
4	(3; 0; 0)	2C ->	(0; 3; 1)
5	(3; 1; 1)	<- 1C	(0; 2; 0)
6	(1; 1; 0)	2M ->	(2; 2; 1)
7	(2; 2; 1)	<- 1M;1C	(1; 1; 0)
8	(0;2; 0)	2M ->	(3; 1; 1)
9	(0; 3; 1)	<- 1C	(3; 0; 0)
10	(0; 1; 0)	2C ->	(3; 2; 1)
11	(0; 2; 1)	<- 1C	(3; 1; 0)
12	(0; 0; 0)	2C ->	(3; 3; 1)

Water Jug Problem

You are given two jugs, a 4- gallon one and a 3- gallon one. Neither have any measuring markers on it. There is a pump that can be used to fill the jugs with water.

How can you get exact 2 gallons of water into the 4- gallon jug?



Initial state: The ordered pair (0; 0).

Actions: Each action is represented in the form " $(x; y) \rightarrow (u; v)$ " where $(x; y)$ represents the state before the application of the action and $(u; v)$ represents the state after the application of the action. The possible actions are given in The Table. The table also specifies the conditions under which the various actions can be applied.

Sl No.	State before action	State after action	Description of operation
1	$(x; y)$ if $x < 4$	$(4; y)$	Fill 4-liter jug
2	$(x; y)$ if $y < 3$	$(x; 3)$	Fill 3-liter jug
3	$(x; y)$ if $x > 0$	$(0; y)$	Empty 4-liter jug on the ground
4	$(x; y)$ if $y > 0$	$(x; 0)$	Empty 3-liter jug on the ground
5	$(x; y)$ if $x + y \geq 4$ and $y > 0$	$(4; y - (4 - x))$	Pour water from 3- liter jug into 4-liter jug until 4-liter jug is full
6	$(x; y)$ if $x + y \geq 3$ and $x > 0$	$(x - (3 - y); 3)$	Pour water from 4- liter jug into 3-liter jug until 3-

			liter jug is full
7	$(x; y)$ if $x + y \leq 4$ and $y > 0$	$(x + y; 0)$	Pour all water from 3- liter jug into 4- liter jug
8	$(x; y)$ if $x + y \leq 3$ and $x > 0$	$(0; x + y)$	Pour all water from 4- liter jug into 3- liter jug

Transition model: The production system given in also specifies the transition model.

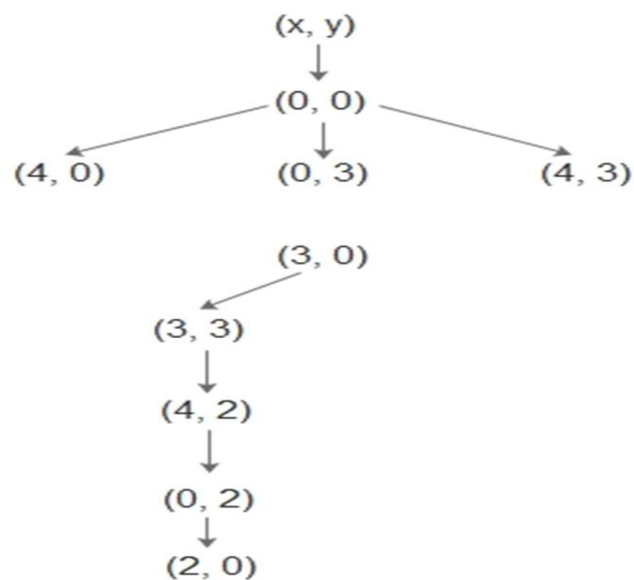
Goal test: The state $(2; n)$ where n is some integer.

Path cost: Number of operations performed.

Solution

One solution to the water jug problem is given below:

$(0; 0) \xrightarrow{\text{Rule 2}} (0; 3) \xrightarrow{\text{Rule 7}} (3; 0) \xrightarrow{\text{Rule 2}} (3; 3) \xrightarrow{\text{Rule 5}} (4; 2) \xrightarrow{\text{Rule 3}} (0; 2) \xrightarrow{\text{Rule 7}} (2; 0)$:



Cryptarithmic puzzle

This is an example for a special type of problem known as “constraint satisfaction problem”. Constraint satisfaction problems are extensively studied and are the subject of intense research in artificial intelligence.

Description

A cryptarithmic puzzle is a mathematical exercise where the digits of some numbers are represented by letters. Each letter represents a unique digit. It will be assumed that the leading digits of numbers are not zero. The problem is to find the digits such that a given mathematical equation is satisfied.

Problem formulation

Using the idea of states the problem can be formulated as follows:

States: Let there be n different letters where $n \leq 10$. A problem state is an ordered n -tuple of digits $(d_1; d_2; \dots; d_n)$ representing the digits to be assigned to the integers.

Initial state: The initial state can be considered as the ordered n -tuple all of whose elements are 0's.

Actions: Increase the value assigned to a letter by 1.

Transition model: Given a state and action, this returns the resulting state.

Goal test: We have reached state $(d_1; d_2; \dots; d_n)$ which satisfies the constraints of the problem.

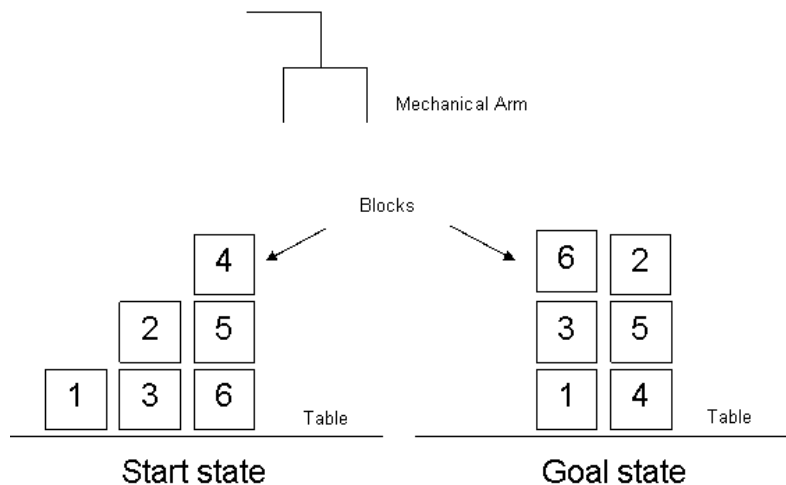
Path cost: Number of actions.

Blocks world problem

Blocks world (or, the world of blocks) is a model domain used in artificial intelligence to explore different approaches to automated reasoning. This model is used to illustrate that a given algorithm can perform planning, or that it is efficient in terms of the number of calculations required to find a solution or in terms of the length of that solution.

Description

There is a table on which some uniform blocks (cubes) are placed. Some blocks may or may not be stacked on other blocks. We have a robot arm to pick up or put down the blocks. The robot arm can move only one block at a time, and no other block should be stacked on top of the block which is to be moved by the robot arm. The problem is to change the configuration of the blocks from a given initial state to a given goal state



States: Configurations of the blocks satisfying the conditions. The configurations can be specified using the following predicates where A and B denote arbitrary blocks:

- ON(A,B) : block A is on block B.
- ONTABLE(A) : block A is on the table.
- CLEAR(A) : block A has nothing on it.
- HOLDING(A) : the arm holds block A.
- ARMEMPTY : the arm holds nothing.

Initial state: The initial configuration of blocks.

Actions:

UNSTACK(A,B) : pick up clear block A from block B and hold it in the arm

STACK(A,B) : place block A held in the arm onto clear block B

PICKUP(A) : lift clear block A with the empty arm

PUTDOWN(A) : place block A held in the arm onto a free space on the table.

Goal state: A certain configuration of the blocks.

Path cost: Number of actions taken to reach the goal state.

Solution

The blocks world problem has always a trivial solution: All blocks not already correctly positioned for the goal state be set off onto the table (one at a time with the mechanical arm), and then reassembled in the proper order on top of any blocks already correctly positioned.

To illustrate the idea consider the blocks world problem specified in Figure. A solution consists of the following 12 moves in that order:

1. UNSTACK(2, 3)
2. PUTDOWN(2)
3. UNSTACK(4, 5)
4. PUTDOWN(4)
5. UNSTACK(5, 6)
6. STACK(5, 4)
7. PICKUP(2)
8. STACK(2, 5)
9. PICKUP(3)
10. STACK(3, 1)
11. PICKUP(6)
12. STACK(6, 3)

Problem Characteristics

1. Is the problem decomposable into a set of independent smaller or easier sub problems?
2. Can solution steps be ignored or at least undone if they prove unwise?
3. Is the problem's universe predictable?
4. Is a good solution to the problem understandable without comparison to all other solutions?
5. Is the desired solution a state of the world or a path to a state?
6. Is a large amount of knowledge absolutely required to solve the problem or is knowledge important only to constrain the search?
7. Can a computer that is simply given the problem return the solution, or will the solution of the problem require interaction between the computer and a person?

1. Is the problem decomposable into a set of independent smaller or easier sub problems?

Consider the problem: Evaluate the integral $\int (x^2 + \sin^2 x) dx$:

We show that this problem is decomposable to smaller sub problems

Since $\int (x^2 + \sin^2 x) dx = \int x^2 dx + \int \sin^2 x dx$: the problem can be decomposed into two simpler sub problems.

_ Problem (i): Evaluate $\int x^2 dx$

_ Problem (ii): Evaluate $\int \sin^2 x dx$

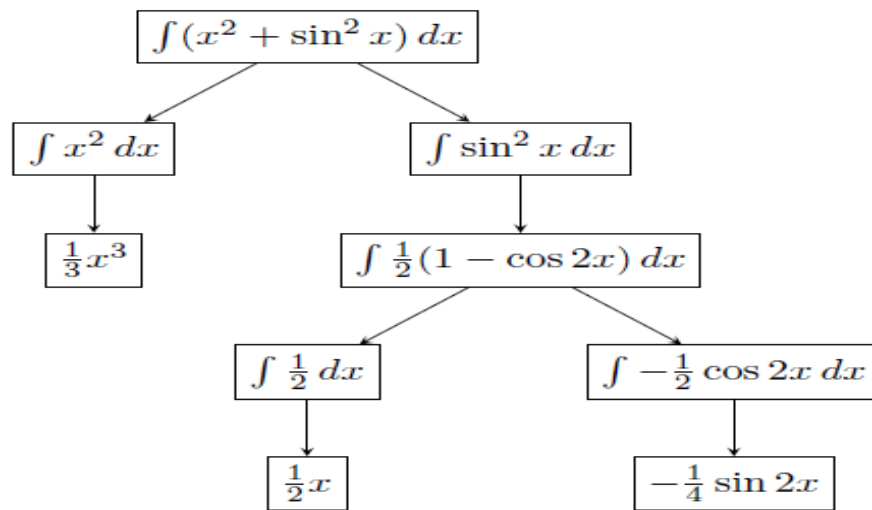


Figure 2.18: Decomposing a problem into smaller subproblems

Combining the solutions of the sub problems we get a solution of the given problem.

2. Solution steps can be ignored or undone.

Example

Suppose our goal is to prove a theorem in mathematics. On the way, we prove some preliminary results, say Result 1, Result 2, Result 3, and then finally we prove the theorem. The steps in the proof look like this:

Result 1

Result 2

Result 3

Theorem

Suppose, we later realize that Result 2 is not actually needed in proving the theorem. Then, we may ignore Result 2 and present the steps of the proof as follows:

Result 1

Result 3

Theorem

In this example, the solution steps can be ***ignored***.

Example 2

Consider the 8-puzzle. The solution involves a sequence of moves. In the process of finding a solution, after some moves, we realize that a certain previous move has been reversed. The previous move can be undone by backtracking the moves. In this example, the solution steps can be ***ignored***.

Example 3

Consider the problem of playing chess. If a chess playing program makes a wrong move, the rules of the game do not permit undoing the move. In this example, the solution steps ***cannot be undone or ignored***.

Ignorable

Ignorable problems can be solved using a simple control structure that never back tracks. Such control structure is easy to implement

Recoverable

Recoverable problems can be solved by more complicated control strategy that sometimes makes mistakes. Back tracking will be necessary to recover from such mistakes.

Irrecoverable

Irrecoverable problems will need to be solved by a system that expends a great deal of effort making each decision since the decision must be final.

3. The problem universe is predictable.

Consider that we are playing with the 8-Puzzle problem. Every time we make a move, we know exactly what will happen. This means that it is possible to plan an entire sequence of moves and be confident what the resulting state will be. We can backtrack to previous moves if they prove wrong.

Suppose we want to play Bridge. We need to make some decisions like which card to play on the first trick. Then we need to make a plan for the entire hand before the first play, but we cannot plan and play with certainty since we cannot know where all the cards are or what the other players will do on their turns. So we have to investigate several plans and use probability of various outcomes to choose best plan for a good score. So, the outcome of this game is very uncertain.

These two games shows us difference between **certain outcome** (e.g. 8-Puzzle) and uncertain outcome (e.g. play bridge) .

In case of **certain outcome**, we can make a plan to generate a sequence of operation that guaranteed to lead to a solution. So, that the outcome is very certain.

In case of **uncertain outcome problems**, we follow the process of plan revision as the plan is carried out and the necessary feedback is provided. The disadvantage is that the planning in this case is often very expensive.

4. There are obvious good solutions without comparison to all other possible solutions.

Example 1

Consider a mathematical problem. In general, there may be many methods for solving the problem. Any method is a good method without comparison to other methods provided it solves the problem. In general, any “any-path problem” is an example of a problem having obvious **good solutions without comparison to other possible solutions**.

Example 2

A “best-path problem” is a problem having no obvious good solutions. The travelling salesman problem is an example for a best-path problem. The travelling salesman problem can be formulated as follows: “Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?”

Example for any path problem

1. Marcus was a man
2. Marcus was a Pompeian
3. Marcus was born in 40 AD
4. All men are mortal
5. All Pompeians died when the volcano erupted in 79AD
6. No mortal lives longer than 150 years
7. It is now 1991 AD

Is Marcus alive?

		Justification
1	Marcus was a man	Axiom 1
4	All men are mortal	Axiom 4
8	Marcus is a mortal	1, 4
3	Marcus was born in 40 AD	Axiom 3
7	It is now 1991 AD	Axiom 7
9	Marcus' age is 1951 years	3, 7
6	No mortal lives longer than 150 years	Axiom 6
10	Marcus is dead	8, 6, 9
	OR	
7	It is now 1991 AD	Axiom 7
5	All Pompeians died in 79AD	Axiom 5
11	All Pompeians died now	7, 5
2	Marcus was a Pompeian	Axiom 2
12	Marcus is dead	11, 2

5. Is the solution a state or a path?

Example 1

In the missionaries and cannibals problem, if we organise the various states in the form of a tree, it can be seen that the solution to the problem is a path connecting the various states

Example 2

In the cryptarithmic puzzle, the solution to the problem is a state of the problem, namely, the state representing the assignment of digits to the letters in the puzzle.

6. What is the role of knowledge?

A lot of knowledge is important only to constrain the search for solution and those for which a lot of knowledge is required to recognize a solution

Example 1

Consider the problem of playing chess. The amount of knowledge required to play chess is very little: just the rules of the game! Additional knowledge about strategies may be used to make intelligent moves!!

Example 2

Consider the problem of scanning daily newspapers to decide which are supporting and which are opposing a political party in an upcoming election. It is obvious that a great deal of knowledge is required to solve this problem.

7. Does the task require Interaction with a Person?

Solitary: - In which the computer is given a problem description and produces an answer with no demand for an explanation of the reasoning process.

Conversational:- In which there is intermediate communication between person and computer either to provide additional assistance to the computer or to provide additional information to the user or both.

Example

Even in a so called “fully automated system” situations constantly arise that call for human intervention. When the machines get thrown off track, or become faulty, experts have to be summoned to step in and troubleshoot the problems.

Problem Classification

When actual problems are examined from the point of view of all the questions, it becomes apparent that there is several board classes into which the problems fall. These classes can each be associated with a generic control strategy that is appropriate for solving the problem.

Production systems

The term “production system” refers to many things. It may refer to a computer program which is used to provide a solution for a problem using a set of rules. It may also refer to a programming language for writing such programs. Further it can also be thought of as a model of computation that can be applied to implement search algorithms or as a model of human problem solving.

A production system (also called production rule system) consists of the following:

1. Set of rules

A set of rules, each consisting of a left side (a pattern) that determines the applicability of the rule and a right side that describes the operation to be performed if the rule is applied.

Briefly, a production rule is a rule of the form “ $C \rightarrow A$ ” which may be interpreted as “given condition C, take action A”. Thus, each rule in the set of rules consists of a left side (a pattern) that determines the applicability of the rule and a right side that describes the operation to be performed if the rule is applied.

2. Knowledge databases

The databases contain whatever information is appropriate for the particular task. Some parts of the database may be permanent.

The database includes a working memory whose elements may pertain only to the solution of the current problem. Working memory contains a description of the current state of the world in the problem-solving process. The description is matched against the conditions of the production rules. When a condition matches, its action is performed.

Actions are designed to alter the contents of working memory.

3. Control strategy

The control strategy includes a specification of the order in which the rules are to be applied and a conflict resolution strategy.

(a) Order of application of rules

There are two methods of execution for a rule system:

- **Forward chaining**

In forward chaining systems, the execution of rules starts from facts in databases and moves towards some goal or conclusion

- **Backward chaining**

In backward chaining, the execution of rules starts from the goals and conclusions and works backwards to see if there is any data from which the goals and conclusions can be derived

(b) Conflict resolution strategy

A system with a large number of rules and facts may result in many rules being true for the same fact; these rules are said to be in conflict. The conflict resolution strategy is a way of resolving the conflicts that manages the execution order of these conflicting rules. Some of the commonly used conflict resolution strategies are listed below:

- Choose arbitrarily.

- Choose the lowest numbered rule.

- Choose the rule containing the largest number of conditions.

- Choose the least recently used rule.

- Choose a rule where the condition is a new fact.

- Choose the rule with the highest priority (weight).

4. Interpreter (rule applier)

The interpreter repeats the following operations: all rules whose conditions are satisfied are found (rule matching), one of them is selected (conflict resolution), and its action is called (rule firing).

The interpreter can be viewed as a "select-execute loop", in which one rule applicable to the current state of the data base is chosen, and then executed. Its action results in a modified data base and the select phase begins again. This cycle is also referred to as a "recognize-act cycle".

Production system: The water jug problem

Problem statement

We have two jugs of capacity 4 liters and 3 liters, and a tap with an endless supply of water. The objective is to obtain 2 liters of water exactly in the 4-liter jug with the minimum steps possible.

Solution

Step 1. Problem formulation as a production system As in section 2.3.8, we represent a state of the problem by an ordered pair $(x; y)$ of integers where x denotes the number of liters of water in the 4-liter jug and y the number of liters of water in the 3-liter jug.

1. Knowledge database

Since there is no information other than that contained in the statement of the problem, the database contains only the working memory. The working memory contains the following two facts:

$\{(0; 0); (2; 0)\}$ where $(0; 0)$ is the initial state and $(2; 0)$ is the goal state.

2. Production rules

Sl No.	Condition	Action	Description
1	(x, y) if $x < 4$	$\rightarrow (4, y)$	Fill 4-liter jug
2	(x, y) if $y < 3$	$\rightarrow (x, 3)$	Fill 3-liter jug
3	(x, y) if $x > 0$	$\rightarrow (0, y)$	Empty 4-liter jug on the ground
4	(x, y) if $y > 0$	$\rightarrow (x, 0)$	Empty 3-liter jug on the ground
5	(x, y) if $x + y \geq 4$ and $y > 0$	$\rightarrow (4, y - (4 - x))$	Pour water from 3-liter jug into 4-liter jug until 4-liter jug is full
6	(x, y) if $x + y \geq 3$ and $x > 0$	$\rightarrow (x - (3 - y), 3)$	Pour water from 4-liter jug into 3-liter jug until 3-liter jug is full
7	(x, y) if $x + y \leq 4$ and $y > 0$	$\rightarrow (x + y, 0)$	Pour all water from 3-liter jug into 4-liter jug
8	(x, y) if $x + y \leq 3$ and $x > 0$	$\rightarrow (0, x + y)$	Pour all water from 4-liter jug into 3-liter jug

3. Conflict resolution strategy

Select and fire the first rule in the conflict set that does not lead to a repeated state.

4. Interpreter

Interpreter is the select-execute loop.

Step 2. Application of the rules

The various steps in the solution procedure obtained by using the forward chaining method is shown in Table

Step No.	Working mem-ory	Current state	Conflict set	Fire rule	Remarks
0	$\{(0, 0), (2, 0)\}$	(0, 0)	–	–	Initial state
1	$\{(0, 0), (2, 0)\}$	(0, 0)	1, 2	1	–
2	$\{(4, 0), (2, 0)\}$	(4, 0)	2, 3, 4, 6	2	–
3	$\{(4, 3), (2, 0)\}$	(4, 3)	3, 4, 5, 6	3	–
4	$\{(0, 3), (2, 0)\}$	(0, 3)	1, 7	7	–
5	$\{(3, 0), (2, 0)\}$	(3, 0)	1, 2, 3, 4, 6, 8	2	–
6	$\{(3, 3), (2, 0)\}$	(3, 3)	1, 3, 4, 5, 6	5	–
7	$\{(4, 2), (2, 0)\}$	(4, 2)	2, 3, 4, 5, 6	3	–
8	$\{(0, 2), (2, 0)\}$	(0, 2)	1, 2, 7	7	–
9	$\{(2, 0), (2, 0)\}$	(2, 0)	–	–	Goal state

3. Conclusion

The operations specified in Table specify a method of getting 2 liters of water in the 4-liter jug. However, this need not necessarily be the optimum solution in the sense that there may be a solution with less number of operations.

Production system characteristics (Classes of production systems)

Production systems are a good way to describe the operations that can be performed in a search for a solution to a problem.

Can production systems, like problems, be described by a set of characteristics that shed light on how they can easily be implemented?

If so, what relationships are there between problem types and the type of production systems best suited to solve the problem?

1. Monotonic and nonmonotonic production systems

A monotonic production system is a production system in which the application of a rule never prevents the later application of another rule that could also have been applied at the time the first rule was selected. A production system which is not monotonic is called a nonmonotonic production system.

2. Partially commutative production systems

A partially commutative production systems is a production system with the property that if the application of a particular sequence of rules transforms state x to state y then any allowable permutation of those rules also transforms state x into state y .

3. Commutative production systems

A production system that is both monotonic and partially commutative is called a commutative production system.

	Monotonic	Non monotonic
Partially commutative	Theorem proving	Robot navigation
Not partially commutative	Chemical synthesis	Bridge

Partially commutative monotonic production systems are useful for solving ignorable problems.

Partially commutative non monotonic production systems are useful for solving in which changes occur but can be reversed and in which order of operation is not critical.

Not partially commutative monotonic production systems are useful for solving problems in which irreversible changes occur

Advantages and disadvantages of production systems

Advantages

1. Provides excellent tools for structuring AI programs.
2. The system is highly modular because individual rules can be added, removed or modified independently.
3. There is separation of knowledge and control.
4. The system uses pattern directed control which is more flexible than algorithmic control.
5. Provides opportunities for heuristic control of the search.
6. Quite helpful in a real-time environment and applications.

Disadvantages

1. There is a lot of inefficiency in production systems. For example, there may be situations where multiple rules get activated during execution and each of these rules may trigger exhaustive searches.
2. It is very difficult to analyze the flow of control within a production system.
3. There is an absence of learning due to a rule-based production system that does not store the result of the problem for future use.
4. The rules in the production system should not have any type of conflict operations. When a new rule is added to the database it should ensure that it does not have any conflict with any existing rule.

Sample questions

1. Describe the 8-puzzle problem.
2. State the missionaries and cannibals problem in AI.
3. What is a production system in AI?
4. Explain the blocks world problem.
5. What is a cryptarithmic puzzle? Illustrate with an example.
6. Briefly explain the concepts of forward chaining and backward chaining in the context of a production system.
7. Define a monotonic production system.

Long answer questions

1. Explain the components of a problem in AI.
2. State the missionaries and cannibals problem and describe its various components as a problem in AI.
3. State the monkey and banana problem and describe its various components as a problem in AI.
4. Briefly explain the characteristics of a problem in AI.
5. Explain the water jugs problem and give the system of production rules for the problem.
6. Describe a production system in AI. What are the merits and demerits of production systems?
7. Define a production rule system and illustrate it with an example.
8. Write a note on the water jug problem using production rules.