CMR College of Engineering & Technology
Kandlakoya (V), Medchal Road, Hyderabad - 501 401. Andhra Pradesh. INDIA
Phone No: 08418 - 200699. Fax No: 08418 - 200240.
E-Mail: principal@cmrcet.org, www.cmrcet.org

CONTENTS

Course Description

Course Objectives

The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence. Emphasis will be placed on the teaching of these fundamentals, not on providing a mastery of specific software tools or programming environments.

1. Gain a historical perspective of AI and its foundations.
2. Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
3. Investigate applications of AI techniques in intelligent agents, expert systems.

Course Outcomes

Students shall be able to

1. Formulate an efficient problem space for a problem expressed in natural language. 2. Apply AI techniques to solve problems of game playing, and logic inference. 3. Deduce first order inference and represent knowledge using appropriate technique 4. Describe various planning and acting for the real-world problems in single agent and multi -agent systems.

5. Explain the concept of uncertainty and probabilistic reasoning.

(A30530) ARTIFICIAL INTELLIGENCE
(Professional Elective-I)

B. Tech (CSE)

L	T	P	C
3	0	0	3

UNIT -I

Problem Solving by Search-I:

Introduction to AI, Intelligent Agents

Problem Solving by Search –II:

Problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth-first search, Uniform cost search, Depth-first search, Iterative deepening Depth-first search, Bidirectional search, Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions, Beyond Classical Search: Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces, Searching with Non-Deterministic Actions, Searching with Partial Observations, Online Search Agents and Unknown Environment .

UNIT –II

Problem Solving by Search-II and Propositional Logic

Adversarial Search:

Games, Optimal Decisions in Games, Alpha–Beta Pruning, Imperfect Real-Time Decisions.

Constraint Satisfaction Problems:

Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems.

Propositional Logic:

Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic

UNIT –III

Logic and Knowledge Representation

First-Order Logic:

Representation, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic:
Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.
Knowledge Representation:
Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

UNIT –IV

Planning
Classical Planning:
Definition of Classical Planning, Algorithms for Planning with State-Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.
Planning and Acting in the Real World:
Time, Schedules, and Resources, Hierarchical Planning, Planning and Acting in Nondeterministic Domains, Multi agent Planning

UNIT –V

Uncertain knowledge and Learning
Uncertainty:
Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use.
Probabilistic Reasoning:
Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.

Learning:
Forms of Learning, Supervised Learning, Learning Decision Trees. Knowledge in Learning: Logical Formulation of Learning, Knowledge in Learning, Explanation-Based Learning, Learning Using Relevance Information, Inductive Logic Programming

TEXT BOOKS:
1. Artificial Intelligence A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson Education

REFERENCE BOOKS:

1. Artificial Intelligence, E. Rich and K. Knight, , 3rd Edition, TMH
2. Artificial Intelligence, Patrick Henny Winston, 3rd Edition, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education.

Course Outcomes

Students shall be able to

1. Formulate an efficient problem space for a problem expressed in natural language.
2. Apply AI techniques to solve problems of game playing, and logic inference.
3. Deduce first order inference and represent knowledge using appropriate technique
4. Describe various palnning and acting for the real-world problems in single agent and multi -agent systems.
5. Explain the concept of uncertainty and probabilistic reasoning.

END

(A30515) COMPUTER NETWORKS LAB

B. Tech (CSE) V Semester	L	T	P	C
	0	0	3	1.5

List of Experiments

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC, CCIP
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijsktra's algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC AUTONOMOUS)
Kandlakoya, Medchal Road, Hyderabad – 501401.

ACADEMIC CALENDAR

Date: 24.06.2023

B.Tech III Year - Academic Year 2023-2024

I Semester

S.No.	Description	Period	Duration
1	Commencement of Class Work	21.08.2023	--------
2	First Spell of Instructions	21.08.2023 to 14.10.2023	8 Weeks
3	*First Mid Examinations*	*16.10.2023 to 21.10.2023*	1 Week
4	Dusara Vacation*	*23.10.2023to 28.10.2023*	1 Week
5	Submission of Mid-I Marks to Exam Branch	30.10.2023	
6	Parent-Teacher Meeting	04.11.2023	
7	Second Spell of Instructions	30.10.2023 to 23.12.2023	8 Weeks
8	*Second MidExaminations*	*25.12.2023 to 30.12.2023*	1 Week
9	Submission of Mid-II Marks to Exam Branch	06.01.2024	
10	Preparations and Practical Examinations	01.01.2024 to 06.01.2024	1 Week
11	*End Semester & Supplementary Examinations*	*08.01.2024 to 20.01.2024*	2 Weeks

II Semester

S.No	Description	Period	Duration
1	Commencement of Class Work	22.01.2024	--------
2	First Spell of Instructions	22.01.2024 to 16.03.2024	8 Weeks
3	*First Mid Examinations*	*18.03.2024 to 23.03.2024*	1 Week
4	Submission of Mid-I Marks to Exam Branch	30.03.2024	
5	Parent-Teacher Meeting	06.04.2024	
6	Second Spell of Instructions	25.03.2024 to 18.05.2024	8 Weeks
7	*Second MidExaminations*	*20.05.2024 to 25.05.2024*	1 Week
8	Submission of Mid-II Marks to Exam Branch	01.06.2024	
9	Preparations and Practical examinations	27.05.2024 to 01.06.2024	1 Week
10	*End Semester & Supplementary Examinations*	*03.06.2024 to 15.06.2024*	*2 Weeks*
11	*Summer vacation*	*17.06.2024 to 29.06.2024*	*2 Weeks*
12	Commencement of Class Work for the next A.Y 2024-2025	01.07.2024	

*Dusara Vacation (Subjected to declaration by JNTUH & TS Govt.)

24/6/2023
Principal
PRINCIPAL
CMR College of Engineering & Technology
(UGC.Autonomous)
Kandlakoya, Medchal Road, Hyderabad, T.S.

Copy submitted to Secretary: for kind information please

Copy to : 1. Deans
2. IQAC
3. All HODs
4. Administrative Officer
5. Accounts Officer
6. Web Portal In charge
7. ERP In Charge
8. Library
9. Student Notice Boards.

CMR College of Engineering & Technology

Department of Mechanical Engineering

SESSION PLANNER

Academic Year: 2023-24
Course Code: A30530
Faculty Name: P.Navya Sree

Semester: V
Course : ARTIFICIAL INTELLIGENCE
Semester Start Date: 21/08/2023

Regulation: R-18
Course Credits: 3
Semester End Date:23/12/2023

Sl. No	Subject Topic Name/ Sub Topic Name	Books	No. of Periods	Cumulative No. of Periods	Planned Week/Date	Completed Date	Delivery Method (White Board/ PPT/ Video links/ URLs /Animation/ Quiz/ Case study/ Model Show case/ 3D Visualization/Mentimeter/ Kahoot/Google classroom/ NPTEL Videos/Pod Cast/ Hands-on/Demos ...etc)
UNIT-I							
1	Problem Solving by Search-I	T1	1	1	21/08/2023 To 27/08/2023		PPT, White Board
2	Intelligent Agents indices,	T1	1	2			PPT, White Board
3	Problem Solving by Search –II: Problem solving agents, Searching for solutions	T1	2	4			PPT, White Board
4	Uninformed Search Strategies	T1	1	5	28/08/2023 To 01/09/2023		PPT, White Board
5	Informed Search Strategies	T1	2	7			PPT, White Board
6	Beyond Classical Search: Hill climbing search, Simulated annealing search	T1	1	8	02/09/2023 To		PPT, White Board

7	Local Search in Continuous Spaces, Searching with Non-Deterministic Actions, searching with partial observations, Online search agents and Unknown environments.	T1	2	10	04/09/2023		PPT, White Board
UNIT-II							
1	**Adversarial Search** : Games, Optimal Decisions in Games, Alpha–Beta Pruning	T1	1	11			PPT, White Board
2	Imperfect Real-Time Decisions. **Constraint Satisfaction Problems:** Defining Constraint Satisfaction Problems	T1	1	12	05/09/2023 To 15/09/2023		PPT, White Board
3	Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs.	T1	2	14			PPT, White Board
4	The Structure of Problems. Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic.	T1	2	16			PPT, White Board
5	**Propositional Theorem Proving:** Inference and proofs, Proof by resolution.	T1	1	17	18/09/2023 To 22/09/2023		PPT, White Board

6	Horn clauses and definite clauses, Forward and backward chaining.	T1	1	18			PPT, White Board
7	Effective Propositional Model Checking, Agents Based on Propositional Logic.	T1	1	19	25/09/2023 To 29/09/2023		PPT, White Board
	UNIT-III						
1	**Logic and Knowledge Representation** **First-Order Logic:** Representation, Syntax and Semantics of First-Order Logic.	T1	1	20	03/10/2023 To 09/10/2023		PPT, White Board
2	Using First-Order Logic, Knowledge Engineering in First-Order Logic.	T1	1	21			PPT, White Board
3	**Inference in First-Order Logic:** Propositional vs. First-Order Inference, Unification and Lifting,	T1	1	22			PPT, White Board
4	Forward Chaining, Backward Chaining, Resolution	T1	1	23	10/10/2023 To 01/11/2023		PPT, White Board

5	**Knowledge Representation:** Ontological Engineering, Categories and Objects, Events.	T1	1	24			PPT, White Board
6	Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.	T1	2	26			PPT, White Board
UNIT-IV							
1	**Planning** **Classical Planning:** Definition of Classical Planning	T1	1	27	02/11/2023 To 06/11/2023		PPT, White Board
2	Algorithms for Planning with State-Space Search, Planning Graphs.	T1	1	28			PPT, White Board
3	Other Classical Planning Approaches, Analysis of Planning approaches.	T1	2	30	08/11/2023 To 13/11/2023		PPT, White Board
4	**Planning and Acting in the Real World:** Time, Schedules, and Resources	T1	1	31			PPT, White Board
5	Hierarchical Planning	T1	1	32			PPT, White Board
6	Planning and Acting in Nondeterministic Domains, Multi agent Planning	T1	1	33			

UNIT-V							
1	**Uncertain knowledge and Learning: Uncertainty:** Acting under Uncertainty, Basic Probability Notation.	T1	1	34	18/11/2023 To 23/11/2023		PPT, White Board
2	Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use	T1	1	35			PPT, White Board
3	**Probabilistic Reasoning:** Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks.	T1	1	36			PPT, White Board
4	Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks	T1	2	37	25/12/2023 To 05/12/2023		PPT, White Board
5	Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.	T1	1	38			PPT, White Board
6	**Learning:** Forms of Learning, Supervised Learning, Learning Decision Trees.	T1	1	39			PPT, White Board
7	Knowledge in Learning: Logical Formulation of Learning, Knowledge in Learning, Explanation based learning	T1	1	40	06/12/2023 To 15/12/2023		PPT, White Board
8	Learning Using Relevance Information, Inductive Logic Programming.	T1	2	41			PPT, White Board

TEXT BOOKS:

1. Artificial Intelligence A Modern Approach, Stuart Russell and Peter Norvig, 3rd Edition, Pearson Education

REFERENCE BOOKS:

1. Artificial Intelligence, E. Rich and K. Knight, , 3rd Edition, TMH

2. 2. Artificial Intelligence, Patrick Henny Winston, 3rd Edition, Pearson Education.
3. 3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. 4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

Name of the Faculty: P.Navya Sree

Signature of Faculty **Dean Academics** **HOD-CSE**

Date: 19-08-2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

Kandlakoya (V), Medchal Road, Hyderabad -501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Sl. No.	Roll Number	Student Name	SEC
1	21H51A0501	BINGI NITHYASRI	A
2	21H51A0503	DASI RASHMIKA	A
3	21H51A0505	GOUNI PAVANI	A
4	21H51A0508	KOMMU VEERENDAR	A
5	21H51A0514	MOHAMMED ABDUL SAMEER	A
6	21H51A0515	MUAAZ MOHAMMED MUNEER	A
7	21H51A0518	PALTHYA SUMAN	A
8	21H51A0519	PAPPULA KARTHIK REDDY	A
9	21H51A0520	POSHETTY VARSHITH	A
10	21H51A0521	RITESH KUMAR	A
11	21H51A0524	TEJAVATH VASANTHA	A
12	21H51A0525	THOTA MAHESHWARI	A
13	21H51A0526	VEERELLI SAIVENKATA REDDY	A
14	21H51A0529	BELKONI ANVESH	A
15	21H51A0533	DASARI AJAY KUMAR	A
16	21H51A0537	GANTA NISHAL	A
17	21H51A0540	KOMMANABOINA ANUSHA	A
18	21H51A0541	LOKOTI SRICHARAN	A
19	21H51A0542	M KAVYA	A
20	21H51A0544	OJAS RAKESH GARPALLIWAR	A
21	21H51A0545	PEDDINTI SAI VARDHAN	A
22	21H51A0547	SATVIKA KARUMUDI	A
23	21H51A0549	THAMMISHETTY SHASHANK	A
24	21H51A0550	TUMMALA VENGAL RAYUDU	A
25	21H51A0551	UMMEDA SHIVA SAI KRISHNA	A
26	21H51A0552	VEMULA PRIYA PRAMIDHA	A
27	21H51A0554	ABHISHEK KUMAR SINGH	A
28	21H51A0555	ALETI ASHWITHA REDDY	A
29	21H51A0556	BATTU VICTOR DINAKAR BABU	A
30	21H51A0559	GANDRATH SRI YAGNA	A
31	21H51A0562	JOGU TARUN TEJA	A
32	21H51A0563	KARRA VINAY REDDY	A
33	21H51A0569	MOHAMMAD FERIA	A
34	21H51A0570	NAGULAPALLY UDAYKIRAN	A
35	21H51A0572	SARVADEY ZANETA	A
36	21H51A0573	SATHYARAM DHANA LAKSHMI	A
37	21H51A0574	SHA SOPNIL JAIN	A
38	21H51A0578	VUPPALA SHLAGHA	A
39	21H51A0582	JYOTHI BALAJI	A
40	21H51A0583	K RITIKA REDDY	A
41	21H51A0584	KOPPULA VENKATA SAI NANDINI	A
42	21H51A0586	M GANESH	A

Sl. No.	Roll Number	Student Name	SEC
43	21H51A0592	NENAVATH SRAVANI RATHOD	A
44	21H51A0595	PAVAN KUMAR	A
45	21H51A0597	ROSHAN TALARI	A
46	21H51A0598	S VARUN	A
47	21H51A05A5	AILENI SATHWIK	A
48	21H51A05A6	AKURATHI RITHVIK SESHAGIRI	A
49	21H51A05A9	BIJJAM SOUMIKA	A
50	21H51A05B0	BODA ASHOK	A
51	21H51A05B6	GOLLAPUDI NITHIN	A
52	21H51A05C1	NALLAKULA KIRANKUMAR	A
53	21H51A05C4	RITVIK PRATHAPANI	A
54	22H55A0501	AILLURI AMARDEEP REDDY	A
55	22H55A0502	BAIROJU SINDHU	A
56	22H55A0503	BODA AVINASH	A
57	22H55A0504	BODA RAHUL SAI KIRAN	A
58	22H55A0505	CHAKILAM BHARAT KUMAR	A
59	22H55A0506	ERLA VENU	A
60	22H55A0507	JONNALA SOWMYA	A
61	22H55A0508	KALE PRABHAS	A
62	22H55A0509	KATKAM MANASWINI	A
63	22H55A0510	KODIDALA KOMALI	A
64	22H55A0511	KONDA MAHIMASRI	A
65	22H55A0512	KONDAPARTHI MANJEERA	A
66	22H55A0513	KUMMARI RAJESH	A
67	22H55A0514	KURUMULA LOKESH	A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Sl. No.	Roll Number	Student Name	SEC
1	21H51A0502	DASARI HARINI	B
2	21H51A0504	GAJULAPALLE SREE LAKSHMI	B
3	21H51A0506	J AKANSH	B
4	21H51A0507	K ZAYD AHMED	B
5	21H51A0509	KURAPATI ESHWAR	B
6	21H51A0510	LAVANGU VAISHNAVI	B
7	21H51A0511	MAHANTHI SAI MANYA SRI	B
8	21H51A0512	MANAS CHHATWAL	B
9	21H51A0513	MANGINA SRI VENKATA SAI	B
10	21H51A0516	NAGIREDDY ANVITHA	B
11	21H51A0517	PADALA ANIL KUMAR	B
12	21H51A0522	SHREYASH SANJEEV KUMAR	B
13	21H51A0523	SIDDAMSHETTI SUMITH	B
14	21H51A0527	AKSHAT KALA	B
15	21H51A0528	ALAVALA KAVYA	B
16	21H51A0530	BENKI JYOTHIKA	B
17	21H51A0531	BENKI VARSHITHA RANI	B
18	21H51A0532	BOLLU HARI CHARHAN	B
19	21H51A0534	DAVULURI SAI SUJAN	B
20	21H51A0535	DESHAPATHI SAHITHI	B
21	21H51A0536	DHULIPALLA VENKATA SAI SIVA	B
22	21H51A0539	KOLAN SAHASRA REDDY	B
23	21H51A0543	MANGA TARAKA RATNA YOSHITH	B
24	21H51A0546	SAPNA TIWARI	B
25	21H51A0548	THAKUR ABHINAV SINGH	B
26	21H51A0553	ABBULA VINUTHNA	B
27	21H51A0557	BUCHENELLI NIKHILESH REDDY	B
28	21H51A0558	DANDA VENKATA SATHWIK REDDY	B
29	21H51A0560	GORINTA RAHULU	B
30	21H51A0561	GUNREDDY AKSHITH REDDY	B
31	21H51A0564	KODURU PRANATHI	B
32	21H51A0565	KONDA VISHAL GOUD	B
33	21H51A0566	KURAKULA SHAILESH	B
34	21H51A0567	MADIRA SAI RISHITHA	B
35	21H51A0568	MANURI CHANDU BABU	B
36	21H51A0571	NIMMALA SAI	B
37	21H51A0575	TUDURU SATHWIK	B
38	21H51A0576	U NAGA MANASWINI	B
39	21H51A0577	VARLA RAMAKRISHNA REDDY	B

Sl. No.	Roll Number	Student Name	SEC
40	21H51A0579	AMBATI ROHITH RAJU	B
41	21H51A0580	BAIRA ANUSHA	B
42	21H51A0581	GUNNALA AKHILA	B
43	21H51A0585	KUDUMULA ANVESH REDDY	B
44	21H51A0587	MANDALOJU VASANTH KUMAR	B
45	21H51A0588	MOHAMMAD ABDUL KALAM	B
46	21H51A0589	MOHAMMED MUDASSIR ALI	B
47	21H51A0590	NALABOLU MOUNIKA	B
48	21H51A0591	NAMPALLY SIDDHARTHA	B
49	21H51A0593	PAMULA BEULAH SUPRAGNYA	B
50	21H51A0594	PANCHAGNULA VINUTNA	B
51	21H51A0596	RAGE DAMODHAR	B
52	21H51A0599	SAI KIRAN B L S	B
53	21H51A05A0	SHESHAVAMATAM SUCHIT PAUL	B
54	21H51A05A1	TEEGALA BHANU TEJA REDDY	B
55	21H51A05A2	VADDI RISHIKA	B
56	21H51A05A3	YADDANAPUDI VISHNU SRIVATSAVA	B
57	21H51A05A4	YELDI ARUN	B
58	21H51A05A7	BAJRANG HARSH SINGH	B
59	21H51A05A8	BASAR SHYAM SUNDER RAO	B
60	21H51A05B1	BUNNI SHARANYA	B
61	21H51A05B2	C J VISHNU PRAKASH	B
62	21H51A05B3	CHIMMULA SHIVA PRASAD REDDY	B
63	21H51A05B4	DOLLA RENUKA	B
64	21H51A05B5	ERUKULA RAJASREE	B
65	21H51A05B7	HARIKA REDDY GANTA	B
66	21H51A05B8	INDUPALLI SHARONSUDHA	B
67	21H51A05B9	MADULAPURAM SAI YASHWANTH RAJ	B
68	21H51A05C0	MALLELA SINDHUJA	B
69	21H51A05C2	RANGU ABHINAV	B
70	21H51A05C3	RAYABARAPU CHATHURYA	B
71	21H51A05C5	SEGU JAYA BALA HARSHAVARDHAN	B
72	21H51A05C8	THATIKONDA AKHILA	B
73	21H51A05C9	VAKALA KAVYA SAI SUMA SRI	B
74	21H51A05D1	ANUJ KUMAR	B
75	21H51A05D2	BACHAWAR VINITHA	B

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
Kandlakoya (V), Medchal Road, Hyderabad -501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Sl. No.	Roll Number	Student Name	SEC
1	21H51A05C6	SOMU KOTESWARA REDDY	C
2	21H51A05C7	SUNKAPAKA JOHN	C
3	21H51A05D0	VALLAMKONDA POOJITHA	C
4	21H51A05D3	BASHAM RAJU	C
5	21H51A05D4	BUSSA TEJASWINI	C
6	21H51A05D5	DADE DINISHA	C
7	21H51A05D6	DEEKONDA SAKETH	C
8	21H51A05E3	MANCHI AKSHAYA	C
9	21H51A05E4	MOHAMMAD ARSHAD NIZAMI	C
10	21H51A05F1	P Y GEETHA MADHURI	C
11	21H51A05F3	SHAIK ILLIYAZ	C
12	21H51A05F5	TUSHAR PUNIA	C
13	21H51A05F8	DODDI SAI PHANI HARI CHANDANA	C
14	21H51A05F9	GADUGULA KALYANI	C
15	21H51A05G2	IYLA SNEHARIKA	C
16	21H51A05G5	KANUGO NESHIT RAJ	C
17	21H51A05H2	PODDUTURI NITHIN REDDY	C
18	21H51A05H8	TADEM RAVITEJA	C
19	21H51A05J8	GUNTHAPALLI MALINI	C
20	21H51A05J9	GURRAM KRISHNA PRASANTH	C
21	21H51A05K3	KODIGANTI SAI KISHORE	C
22	21H51A05K8	SEELAMSETTY PRASANNA GAYATHRI	C
23	21H51A05L1	SRIRAM NAGARAJU	C
24	21H51A05L7	YALLA TEJASWIK REDDY	C
25	21H51A05M0	CHILUKA SAI KARTHIK	C
26	21H51A05M1	DAMARLA HEMAVATHI	C
27	21H51A05M4	GIRAVENA ARYA	C
28	21H51A05M9	MOKIRALA JHANSI	C
29	21H51A05N1	NEELA SAI ADITYA	C
30	21H51A05N3	POTRU SAI NITISH	C
31	21H51A05N4	PRAHARSHITHA SURAGONI	C
32	21H51A05N5	PULI PRANEETH GOUD	C
33	21H51A05P0	TALOORI PRABHU KIRAN	C
34	21H51A05P2	VAVILLA RAVITEJA	C
35	21H51A05P4	ALLURI SAI SATHWIK REDDY	C
36	21H51A05P5	ANDE AJAY	C
37	21H51A05P7	BESTHA NANDA KISHORE	C
38	21H51A05P8	CHAVATAPALLI MUKUNDA SRI HASINI	C
39	21H51A05P9	CHEPYALA SATHWIK REDDY	C

Sl. No.	Roll Number	Student Name	SEC
40	21H51A05Q1	DAGGULA PRASHANTH	C
41	21H51A05Q2	GAJULA NAVANEETH	C
42	21H51A05Q3	GUDAPATI NITHIN KUMAR	C
43	21H51A05R3	PINAPATI ABHISHEK	C
44	21H51A05R4	RACHAMALLA SAI UJITHA REDDY	C
45	21H51A05R5	SATTU RAKESH	C
46	21H51A05R6	SHREYA M	C
47	21H51A05R7	YERAVELLI RUCHITHA	C
48	22H55A0515	M. SAI RANJITH REDDY	C
49	22H55A0516	MAHATHI DESAI	C
50	22H55A0517	MD TOWHEED	C
51	22H55A0518	MOHAMMED HANEF	C
52	22H55A0519	NAGARAM SHIVA CHAND	C
53	22H55A0520	NARGE CHARANETEJA	C
54	22H55A0521	NEELAM RAMYA SARI	C
55	22H55A0522	PANDAV SONIA	C
56	22H55A0523	PATHLAVATH SUNITHA	C
57	22H55A0524	POTTIPALLY DEEPIKA	C
58	22H55A0525	PULIGANTI MAHENDAR	C
59	22H55A0526	SARDESHI PRAVEEN KUMAR	C
60	22H55A0527	VISLAVATH ANITHA	C

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Sl. No.	Roll Number	Student Name	SEC
1	21H51A05D7	DHUDURI SATHVIKA	D
2	21H51A05D8	GAMPALA SRI DURGA PRABHATH	D
3	21H51A05D9	GUNDLA VAMSHIDHAR	D
4	21H51A05E0	KASANAGOTTU AMULYA	D
5	21H51A05E1	KOTHA VAISHNAVI	D
6	21H51A05E2	KUMBALA ABHILASH REDDY	D
7	21H51A05E6	NAKKALA KEERTHANA	D
8	21H51A05E7	NEELAM BHARATH KUMAR	D
9	21H51A05E8	NEERUDI HARIPRASAD	D
10	21H51A05E9	ODURI VEERAMANIKANTA	D
11	21H51A05F0	OM GUPTA	D
12	21H51A05F2	ROHAN SACHIN RAKHE	D
13	21H51A05F4	SHAIK TASNIM	D
14	21H51A05F6	YARRAMSETTI MADHU VENKATA	D
15	21H51A05F7	BABBI THAPA	D
16	21H51A05G0	GUDIPALLY SAI SANJAY	D
17	21H51A05G1	GUNNA VINAY KUMAR REDDY	D
18	21H51A05G3	K SRI HARINI	D
19	21H51A05G4	KANDI SWETHA	D
20	21H51A05G6	KHANDESH THANU SRI	D
21	21H51A05G7	MAMIDI VENU GOPAL	D
22	21H51A05G8	MARAGONI KARTHIKEYA	D
23	21H51A05G9	NALIMELA JITHIN REDDY	D
24	21H51A05H1	PATRAYADI RAVI	D
25	21H51A05H3	POTHARAJU SAI KIRAN	D
26	21H51A05H5	SHERIKAR RAHUL	D
27	21H51A05H6	SOMARAJUPALLI THEJASWI	D
28	21H51A05H7	SUDAM SHIVA	D
29	21H51A05H9	THALLAM GEETHAN	D
30	21H51A05J0	TODUPUNURI SHAI PRIYA	D
31	21H51A05J1	TUMMALA SAHITH	D
32	21H51A05J2	VIJAYAGIRI AMULYA	D
33	21H51A05J3	ABHAY PRATAP SINGH	D
34	21H51A05J4	AYEMON ZEBA	D
35	21H51A05J5	BONDALA SRINATH	D
36	21H51A05J6	DODDAPANENI MEGHAN CHOWDARY	D
37	21H51A05J7	GORANTI SANTHU SATHWIK	D
38	21H51A05K0	KACHIREDDY JAYASREE	D
39	21H51A05K1	KAJA SANJEEV KUMAR	D

Sl. No.	Roll Number	Student Name	SEC
40	21H51A05K2	KANTU ANANTHKUMAR	D
41	21H51A05K4	KONDETI VIKRAMREDDY	D
42	21H51A05K5	KRITIKA KHATRI	D
43	21H51A05K6	NITYANANDAYYA MATHPATHI	D
44	21H51A05K9	SHANIGALA VISHNU	D
45	21H51A05L0	SINDEY ABHIGNA	D
46	21H51A05L2	SUMESH	D
47	21H51A05L3	TANNIRU MAHESH	D
48	21H51A05L4	TUSYAA SREERALA	D
49	21H51A05L5	UNI SAILESH	D
50	21H51A05L6	VAGUAMRI SRINANDHAN	D
51	21H51A05L8	BEHARA SURAJ	D
52	21H51A05L9	BHAKE SHASHANK	D
53	21H51A05M2	DIVYA GAUTAM	D
54	21H51A05M3	GANGASANI SHANKARSHAN	D
55	21H51A05M5	GUMMIREDDY SAINATH REDDY	D
56	21H51A05M6	KALLURI THANMAI	D
57	21H51A05M7	KATRAVATH MANJULA	D
58	21H51A05M8	MOHAMMED SAMEER ALI	D
59	21H51A05N0	NANCHARLA SAI AKSHITHA	D
60	21H51A05N2	OLIGE RANI	D
61	21H51A05N6	SAKKERLA RAJ KUMAR	D
62	21H51A05N7	SALENDRA MANOJ KUMAR	D
63	21H51A05N8	SHAIK JAVED	D
64	21H51A05N9	SHRIYA MALANI	D
65	21H51A05P1	VASURI VINAY KUMAR	D
66	21H51A05P3	VITTAPUR BINNU REDDY	D
67	21H51A05P6	BANOTHU DALI HIMASRI	D
68	21H51A05Q0	D GAYATHRI	D
69	21H51A05Q4	GUDIPUDI DHEERAJ	D
70	21H51A05Q5	GURRAM SRIKANTH	D
71	21H51A05Q6	KALVAKUNTA CHANDRASHEKAR	D
72	21H51A05Q7	KAPU HARSHA VARDAN REDDY	D
73	21H51A05Q8	KOTTE MOUNIKA	D
74	21H51A05Q9	MANDA VIGHNESHWARA REDDY	D
75	21H51A05R0	MANDHUMULA DEEPAK	D
76	21H51A05R1	PEDDI PRAVALIKA REDDY	D
77	21H51A05R2	PENDEM YOGITHA	D
78	21H51A05R8	YESUGARI ADHARSH	D

CMR College of Engineering & Technology

(UGC AUTONOMOUS)

Kandlakoya, Medchal Road - 501401

CMR

Department of Computer Science and Engineering

MID-I MARKS LIST

Class : III B.Tech. I SEM CSE SECTION-A A.Y.2023-24

SUBJECT : Artificial Intelligence

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
1	21H51A0501	BINGI NITHYASRI	5	18	23
2	21H51A0503	DASI RASHMIKA	5	18	23
3	21H51A0505	GOUNI PAVANI	5	21	26
4	21H51A0508	KOMMU VEERENDAR	5	18	23
5	21H51A0514	MOHAMMED ABDUL SAMEER	5	02	07
6	21H51A0515	MUAAZ MOHAMMED MUNEER	5	13	18
7	21H51A0518	PALTHYA SUMAN	5	16	21
8	21H51A0519	PAPPULA KARTHIK REDDY	A	08	08
9	21H51A0520	POSHETTY VARSHITH	5	05	10
10	21H51A0521	RITESH KUMAR	5	04	09
11	21H51A0524	TEJAVATH VASANTHA	5	23	28
12	21H51A0525	THOTA MAHESHWARI	5	15	20
A 13	21H51A0526	VEERELLI SAIVENKATA REDDY	5	A	05
14	21H51A0529	BELKONI ANVESH	5	22	27
15	21H51A0533	DASARI AJAY KUMAR	5	16	21
16	21H51A0537	GANTA NISHAL	5	08	13
17	21H51A0540	KOMMANABOINA ANUSHA	5	17	22
18	21H51A0541	LOKOTI SRICHARAN	5	02	07
19	21H51A0542	M KAVYA	5	10	15
20	21H51A0544	OJAS RAKESH GARPALLIWAR	5	11	16
21	21H51A0545	PEDDINTI SAI VARDHAN	3	05	08
22	21H51A0547	SATVIKA KARUMUDI	5	16	21
23	21H51A0549	THAMMISHETTY SHASHANK	4	09	13
24	21H51A0550	TUMMALA VENGAL RAYUDU	5	02	07
A 25	21H51A0551	UMMEDA SHIVA SAI KRISHNA	A	~~A~~ 20	~~A~~ 20 Nay
26	21H51A0552	VEMULA PRIYA PRAMIDHA	5	21	26
27	21H51A0554	ABHISHEK KUMAR SINGH	5	20	25
28	21H51A0555	ALETI ASHWITHA REDDY	5	19	24
29	21H51A0556	BATTU VICTOR DINAKAR BABU	5	14	19
30	21H51A0559	GANDRATH SRI YAGNA	A	07	07
31	21H51A0562	JOGU TARUN TEJA	5	10	15
32	21H51A0563	KARRA VINAY REDDY	5	06	11
33	21H51A0569	MOHAMMAD FERIA	5	14	19
34	21H51A0570	NAGULAPALLY UDAYKIRAN	5	04	09
35	21H51A0572	SARVADEY ZANETA	5	17	22
36	21H51A0573	SATHYARAM DHANA LAKSHMI	5	21	26
37	21H51A0574	SHA SOPNIL JAIN	5	20	25
38	21H51A0578	VUPPALA SHLAGHA	5	17	22
39	21H51A0582	JYOTHI BALAJI	5	0	05
40	21H51A0583	K RITIKA REDDY	5	05	10

4 → Ab

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)	
A~ 41	21H51A0584	KOPPULA VENKATA SAI NANDINI	5	~~A~~ 04	~~05~~ 09	Navy
42	21H51A0586	M GANESH	5	13	18	
43	21H51A0592	NENAVATH SRAVANI RATHOD	5	20	25	
A~ 44	21H51A0595	PAVAN KUMAR	A	~~A~~ 16	~~A~~ 16	Navy
45	21H51A0597	ROSHAN TALARI	5	22	27	
46	21H51A0598	S VARUN	5	17	22	
47	21H51A05A5	AILENI SATHWIK	5	18	23	
48	21H51A05A6	AKURATHI RITHVIK SESHAGIRI	5	06	11	
49	21H51A05A9	BIJJAM SOUMIKA	3	05	08	
50	21H51A05B0	BODA ASHOK	5	06	11	
51	21H51A05B6	GOLLAPUDI NITHIN	4	04	08	
52	21H51A05C1	NALLAKULA KIRANKUMAR	5	03	08	
53	21H51A05C4	RITVIK PRATHAPANI	5	09	14	
54	22H55A0501	AILLURI AMARDEEP REDDY	5	14	19	
55	22H55A0502	BAIROJU SINDHU	5	16	21	
56	22H55A0503	BODA AVINASH	5	11	16	
57	22H55A0504	BODA RAHUL SAI KIRAN	A	05	05	
58	22H55A0505	CHAKILAM BHARAT KUMAR	5	23	28	
59	22H55A0506	ERLA VENU	5	14	19	
60	22H55A0507	JONNALA SOWMYA	5	19	24	
61	22H55A0508	KALE PRABHAS	5	17	22	
62	22H55A0509	KATKAM MANASWINI	5	18	23	
63	22H55A0510	KODIDALA KOMALI	5	17	22	
64	22H55A0511	KONDA MAHIMASRI	5	20	25	
65	22H55A0512	KONDAPARTHI MANJEERA	5	22	27	
66	22H55A0513	KUMMARI RAJESH	5	09	14	
67	22H55A0514	KURUMULA LOKESH	5	15	20	

Name&Signature of the Faculty :	P. Navya Sree P. Navya
Department :	CSE
Mobile No :	8555059731.

HOD/CSE

CMR College of Engineering & Technology

(UGC AUTONOMOUS)

Kandlakoya, Medchal Road - 501401

Department of Computer Science and Engineering

MID-I MARKS LIST

Class : III B.Tech. I SEM CSE SECTION-B A.Y.2023-24

SUBJECT : Artificial Intelligence

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
1	21H51A0502	DASARI HARINI	05	23	28
2	21H51A0504	GAJULAPALLE SREE LAKSHMI	05	24	29
3	21H51A0506	J AKANSH	05	16	21
4	21H51A0507	K ZAYD AHMED	05	15	20
5	21H51A0509	KURAPATI ESHWAR	05	25	30
6	21H51A0510	LAVANGU VAISHNAVI	05	21	26
7	21H51A0511	MAHANTHI SAI MANYA SRI	05	24	29
8	21H51A0512	MANAS CHHATWAL	05	22	27
9	21H51A0513	MANGINA SRI VENKATA SAI	05	22	27
10	21H51A0516	NAGIREDDY ANVITHA	05	24	29
11	21H51A0517	PADALA ANIL KUMAR	05	20	25
12	21H51A0522	SHREYASH SANJEEV KUMAR	05	20	25
13	21H51A0523	SIDDAMSHETTI SUMITH	05	15	20
14	21H51A0527	AKSHAT KALA	05	17	22
15	21H51A0528	ALAVALA KAVYA	05	24	29
16	21H51A0530	BENKI JYOTHIKA	05	21	26
17	21H51A0531	BENKI VARSHITHA RANI	05	23	28
18	21H51A0532	BOLLU HARI CHARHAN	05	18	23
19	21H51A0534	DAVULURI SAI SUJAN	05	19	24
20	21H51A0535	DESHAPATHI SAHITHI	05	24	29
21	21H51A0536	DHULIPALLA VENKATA SAI SIVA	05	23	28
22	21H51A0539	KOLAN SAHASRA REDDY	05	22	27
23	21H51A0543	MANGA TARAKA RATNA YOSHITH	05	18	23
24	21H51A0546	SAPNA TIWARI	05	19	24
25	21H51A0548	THAKUR ABHINAV SINGH	05	23	28
26	21H51A0553	ABBULA VINUTHNA	05	21	26
27	21H51A0557	BUCHENELLI NIKHILESH REDDY	05	19	24
28	21H51A0558	DANDA VENKATA SATHWIK REDDY	05	22	27
29	21H51A0560	GORINTA RAHULU	05	22	27
30	21H51A0561	GUNREDDY AKSHITH REDDY	05	21	26
31	21H51A0564	KODURU PRANATHI	05	23	28
32	21H51A0565	KONDA VISHAL GOUD	05	22	27
33	21H51A0566	KURAKULA SHAILESH	05	19	24
34	21H51A0567	MADIRA SAI RISHITHA	05	24	29
35	21H51A0568	MANURI CHANDU BABU	05	21	26
36	21H51A0571	NIMMALA SAI	05	20	25
37	21H51A0575	TUDURU SATHWIK	05	19	24
38	21H51A0576	U NAGA MANASWINI	05	18	23
39	21H51A0577	VARLA RAMAKRISHNA REDDY	05	20	25
40	21H51A0579	AMBATI ROHITH RAJU	05	21	26

0 → AB

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
41	21H51A0580	BAIRA ANUSHA	05	23	28
42	21H51A0581	GUNNALA AKHILA	05	25	30
43	21H51A0585	KUDUMULA ANVESH REDDY	05	25	30
44	21H51A0587	MANDALOJU VASANTH KUMAR	05	19	24
45	21H51A0588	MOHAMMAD ABDUL KALAM	05	23	28
46	21H51A0589	MOHAMMED MUDASSIR ALI	05	19	24
47	21H51A0590	NALABOLU MOUNIKA	05	22	27
48	21H51A0591	NAMPALLY SIDDHARTHA	05	25	30
49	21H51A0593	PAMULA BEULAH SUPRAGNYA	05	15	20
50	21H51A0594	PANCHAGNULA VINUTNA	05	25	30
51	21H51A0596	RAGE DAMODHAR	04	21	25
52	21H51A0599	SAI KIRAN B L S	04	11	15
53	21H51A05A0	SHESHAVAMATAM SUCHIT PAUL	05	20	25
54	21H51A05A1	TEEGALA BHANU TEJA REDDY	05	20	25
55	21H51A05A2	VADDI RISHIKA	05	25	30
56	21H51A05A3	YADDANAPUDI VISHNU SRIVATSAVA	05	23	28
57	21H51A05A4	YELDI ARUN	05	22	27
58	21H51A05A7	BAJRANG HARSH SINGH	05	21	26
59	21H51A05A8	BASAR SHYAM SUNDER RAO	05	22	27
60	21H51A05B1	BUNNI SHARANYA	05	25	30
61	21H51A05B2	C J VISHNU PRAKASH	05	17	22
62	21H51A05B3	CHIMMULA SHIVA PRASAD REDDY	05	24	29
63	21H51A05B4	DOLLA RENUKA	05	23	28
64	21H51A05B5	ERUKULA RAJASREE	05	21	26
65	21H51A05B7	HARIKA REDDY GANTA	05	20	25
66	21H51A05B8	INDUPALLI SHARONSUDHA	05	22	27
67	21H51A05B9	MADULAPURAM SAI YASHWANTH RAJ	05	20	25
68	21H51A05C0	MALLELA SINDHUJA	05	24	29
69	21H51A05C2	RANGU ABHINAV	05	22	27
70	21H51A05C3	RAYABARAPU CHATHURYA	05	23	28
71	21H51A05C5	SEGU JAYA BALA HARSHAVARDHAN	05	22	27
72	21H51A05C8	THATIKONDA AKHILA	05	22	27
73	21H51A05C9	VAKALA KAVYA SAI SUMA SRI	05	22	27
74	21H51A05D1	ANUJ KUMAR	05	21	26
75	21H51A05D2	BACHAWAR VINITHA	05	23	28

Name&Signature of the Faculty :	P. Navya Sree P. Navya
Department :	CSE
Mobile No :	8555059731

HOD/CSE

CMR College of Engineering & Technology

(UGC AUTONOMOUS)

Kandlakoya, Medchal Road - 501401

Department of Computer Science and Engineering

MID-I MARKS LIST

Class : III B.Tech. I SEM CSE SECTION-C A.Y.2023-24

SUBJECT : Artificial Intelligence

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
1	21H51A05C6	SOMU KOTESWARA REDDY	5	21	26
2	21H51A05C7	SUNKAPAKA JOHN	5	19	24
3	21H51A05D0	VALLAMKONDA POOJITHA	5	22	27
4	21H51A05D3	BASHAM RAJU	5	19	24
5	21H51A05D4	BUSSA TEJASWINI	5	15	20
6	21H51A05D5	DADE DINISHA	5	16	21
7	21H51A05D6	DEEKONDA SAKETH	AB	13	13
8	21H51A05E3	MANCHI AKSHAYA	5	21	26
9	21H51A05E4	MOHAMMAD ARSHAD NIZAMI	5	11	16
10	21H51A05F1	P Y GEETHA MADHURI	5	18	23
11	21H51A05F3	SHAIK ILLIYAZ	←	AB	→
12	21H51A05F5	TUSHAR PUNIA	5	22	27
13	21H51A05F8	DODDI SAI PHANI HARI CHANDANA	5	20	25
14	21H51A05F9	GADUGULA KALYANI	5	22	27
15	21H51A05G2	IYLA SNEHARIKA	5	20	25
16	21H51A05G5	KANUGO NESHIT RAJ	5	17	22
17	21H51A05H2	PODDUTURI NITHIN REDDY	5	7	12
18	21H51A05H8	TADEM RAVITEJA	5	15	20
19	21H51A05J8	GUNTHAPALLI MALINI	5	12	17
20	21H51A05J9	GURRAM KRISHNA PRASANTH	←	~~AB~~ 06	06 → Nay
21	21H51A05K3	KODIGANTI SAI KISHORE	← AB	~~AB~~ AB 18	18 → Nay
22	21H51A05K8	SEELAMSETTY PRASANNA GAYATHRI	5	18	23
23	21H51A05L1	SRIRAM NAGARAJU	5	18	23
24	21H51A05L7	YALLA TEJASWIK REDDY	5	14	19
25	21H51A05L8	BEHARA SURAJ	5	22	27
26	21H51A05M0	CHILUKA SAI KARTHIK	5	16	21
27	21H51A05M1	DAMARLA HEMAVATHI	5	16	21
28	21H51A05M4	GIRAVENA ARYA	5	9	14
29	21H51A05M9	MOKIRALA JHANSI	5	22	27
30	21H51A05N1	NEELA SAI ADITYA	5	11	16
31	21H51A05N3	POTRU SAI NITISH	5	11	16
32	21H51A05N4	PRAHARSHITHA SURAGONI	5	21	26
33	21H51A05N5	PULI PRANEETH GOUD	5	14	19
34	21H51A05P0	TALOORI PRABHU KIRAN	5	18	23
35	21H51A05P2	VAVILLA RAVITEJA	5	24	29
36	21H51A05P4	ALLURI SAI SATHWIK REDDY	5	19	24
37	21H51A05P5	ANDE AJAY	5	← 10 ~~[illegible]~~ →	~~05~~ 15 Nav
38	21H51A05P7	BESTHA NANDA KISHORE	5	20	25
39	21H51A05P8	CHAVATAPALLI MUKUNDA SRI HASINI	5	17	22
40	21H51A05P9	CHEPYALA SATHWIK REDDY	5	20	25

3 → Ab

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
41	21H51A05Q1	DAGGULA PRASHANTH	5	3	8
42	21H51A05Q2	GAJULA NAVANEETH	5	17	22
43	21H51A05Q3	GUDAPATI NITHIN KUMAR	5	4	9
44	21H51A05R3	PINAPATI ABHISHEK	5	17	22
45	21H51A05R4	RACHAMALLA SAI UJITHA REDDY	5	16	21
46	21H51A05R5	SATTU RAKESH	5	20	25
47	21H51A05R6	SHREYA M	5	15	20
48	21H51A05R7	YERAVELLI RUCHITHA	5	17	22
49	22H55A0515	M. SAI RANJITH REDDY	5	20	25
50	22H55A0516	MAHATHI DESAI	5	15	20
51	22H55A0517	MD TOWHEED	AB	13	13
52	22H55A0518	MOHAMMED HANEF	5	14	19
53	22H55A0519	NAGARAM SHIVA CHAND	5	22	27
54	22H55A0520	NARGE CHARANETEJA	5	22	27
55	22H55A0521	NEELAM RAMYA SARI	5	19	24
56	22H55A0522	PANDAV SONIA	5	19	24
57	22H55A0523	PATHLAVATH SUNITHA	5	18	23
58	22H55A0524	POTTIPALLY DEEPIKA	5	15	20
59	22H55A0525	PULIGANTI MAHENDAR	5	17	22
60	22H55A0526	SARDESHI PRAVEEN KUMAR	5	20	25
61	22H55A0527	VISLAVATH ANITHA	5	21	26

Name&Signature of the Faculty :	P. Adarsana (signature)
Department :	CSE
Mobile No :	9346465781.

(signature)
HOD/CSE

CMR College of Engineering & Technology

(UGC AUTONOMOUS)

Kandlakoya, Medchal Road - 501401

Department of Computer Science and Engineering

MID-I MARKS LIST

Class : III B.Tech. I SEM CSE SECTION-D A.Y.2023-24

SUBJECT : ARTIFICIAL INTELLIGENCE

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
1	21H51A05D7	DHUDURI SATHVIKA	5	20	25
2	21H51A05D8	GAMPALA SRI DURGA PRABHATH	5	21	26
3	21H51A05D9	GUNDLA VAMSHIDHAR	5	17	22
4	21H51A05E0	KASANAGOTTU AMULYA	5	20	25
5	21H51A05E1	KOTHA VAISHNAVI	5	22	27
6	21H51A05E2	KUMBALA ABHILASH REDDY	5	15	20
7	21H51A05E6	NAKKALA KEERTHANA	5	21	26
8	21H51A05E7	NEELAM BHARATH KUMAR	5	21	26
9	21H51A05E8	NEERUDI HARIPRASAD	5	21	26
10	21H51A05E9	ODURI VEERAMANIKANTA	5	15	20
11	21H51A05F0	OM GUPTA	5	17	22
12	21H51A05F2	ROHAN SACHIN RAKHE	5	16	21
13	21H51A05F4	SHAIK TASNIM	5	19	24
14	21H51A05F6	YARRAMSETTI MADHU VENKATA	5	17	22
15	21H51A05F7	BABBI THAPA	5	20	25
16	21H51A05G0	GUDIPALLY SAI SANJAY	5	18	23
17	21H51A05G1	GUNNA VINAY KUMAR REDDY	5	22	27
18	21H51A05G3	K SRI HARINI	5	20	25
19	21H51A05G4	KANDI SWETHA	5	17	22
20	21H51A05G6	KHANDESH THANU SRI	5	17	22
21	21H51A05G7	MAMIDI VENU GOPAL	5	14	19
22	21H51A05G8	MARAGONI KARTHIKEYA	5	21	26
23	21H51A05G9	NALIMELA JITHIN REDDY	5	16	21
24	21H51A05H1	PATRAYADI RAVI	5	19	24
25	21H51A05H3	POTHARAJU SAI KIRAN	5	19	24
26	21H51A05H5	SHERIKAR RAHUL	5	18	23
27	21H51A05H6	SOMARAJUPALLI THEJASWI	5	16	21
28	21H51A05H7	SUDAM SHIVA	5	16	21
29	21H51A05H9	THALLAM GEETHAN	5	13	18
30	21H51A05J0	TODUPUNURI SHALPRIYA	5	23	28
31	21H51A05J1	TUMMALA SAHITH	4	12	16
32	21H51A05J2	VIJAYAGIRI AMULYA	5	21	26
33	21H51A05J3	ABHAY PRATAP SINGH	5	19	24
34	21H51A05J4	AYEMON ZEBA	5	17	22
35	21H51A05J5	BONDALA SRINATH	5	17	22
36	21H51A05J6	DODDAPANENI MEGHAN CHOWDARY	5	17	22
37	21H51A05J7	GORANTI SANTHU SATHWIK	5	19	24
38	21H51A05K0	KACHIREDDY JAYASREE	5	21	26
39	21H51A05K1	KAJA SANJEEV KUMAR	5	21	26
40	21H51A05K2	KANTU ANANTHKUMAR	5	15	20

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
41	21H51A05K4	KONDETI VIKRAMREDDY	5	15	20
42	21H51A05K5	KRITIKA KHATRI	5	17	22
43	21H51A05K6	NITYANANDAYYA MATHPATHI	5	14	19
44	21H51A05K9	SHANIGALA VISHNU	5	17	22
45	21H51A05L0	SINDEY ABHIGNA	5	20	25
46	21H51A05L2	SUMESH	5	18	23
47	21H51A05L3	TANNIRU MAHESH	5	14	19
48	21H51A05L4	TUSYAA SREERALA	5	18	23
49	21H51A05L5	UNI SAILESH	5	18	23
50	21H51A05L6	VAGUAMRI SRINANDHAN	5	18	23
51	21H51A05L9	BHAKE SHASHANK	5	18	23
52	21H51A05M2	DIVYA GAUTAM	5	19	24
53	21H51A05M3	GANGASANI SHANKARSHAN	5	20	25
54	21H51A05M5	GUMMIREDDY SAINATH REDDY	5	17	22
55	21H51A05M6	KALLURI THANMAI	5	17	22
56	21H51A05M7	KATRAVATH MANJULA	5	19	24
57	21H51A05M8	MOHAMMED SAMEER ALI	5	13	18
58	21H51A05N0	NANCHARLA SAI AKSHITHA	5	16	21
59	21H51A05N2	OLIGE RANI	5	18	23
60	21H51A05N6	SAKKERLA RAJ KUMAR	5	10	15
61	21H51A05N7	SALENDRA MANOJ KUMAR	5	8	13
62	21H51A05N8	SHAIK JAVED	5	17	22
63	21H51A05N9	SHRIYA MALANI	5	17	22
64	21H51A05P1	VASURI VINAY KUMAR	5	18	23
65	21H51A05P3	VITTAPUR BINNU REDDY	5	19	24
66	21H51A05P6	BANOTHU DALI HIMASRI	5	18	23
67	21H51A05Q0	D GAYATHRI	5	21	26
68	21H51A05Q4	GUDIPUDI DHEERAJ	5	14	19
69	21H51A05Q5	GURRAM SRIKANTH	5	16	21
70	21H51A05Q6	KALVAKUNTA CHANDRASHEKAR	5	16	21
71	21H51A05Q7	KAPU HARSHA VARDAN REDDY	5	20	25
72	21H51A05Q8	KOTTE MOUNIKA	5	19	24
73	21H51A05Q9	MANDA VIGHNESHWARA REDDY	5	13	18
74	21H51A05R0	MANDHUMULA DEEPAK	5	17	22
A — 75	21H51A05R1	PEDDI PRAVALIKA REDDY	5	15	20
76	21H51A05R2	PENDEM YOGITHA	5	19	24
77	21H51A05R8	YESUGARI ADHARSH	5	18	23

Name&Signature of the Faculty :	T. Adarsana (Adarsana)
Designation:	Asst. prof
Department :	CSE
Mobile No :	9346465781.

HOD/CSE

CMR College of Engineering & Technology

(UGC AUTONOMOUS)

Kandlakoya , Medchal Road - 501401

Department of Computer Science and Engineering

MID-II MARKS LIST

Class : III B.Tech. I SEM CSE SECTION-A A.Y.2023-24

SUBJECT : Artificial Intelligence.

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
1	21H51A0501	BINGI NITHYASRI	AB	07	07
2	21H51A0503	DASI RASHMIKA	5	09	14
3	21H51A0505	GOUNI PAVANI	5	10	15
4	21H51A0508	KOMMU VEERENDAR	5	12	17
5	21H51A0514	MOHAMMED ABDUL SAMEER	5	11	16
6	21H51A0515	MUAAZ MOHAMMED MUNEER	5	11	16
7	21H51A0518	PALTHYA SUMAN	5	15	20
8	21H51A0519	PAPPULA KARTHIK REDDY	AB	07	07
9	21H51A0520	POSHETTY VARSHITH	AB	04	04
10	21H51A0521	RITESH KUMAR	5	07	12
11	21H51A0524	TEJAVATH VASANTHA	5	13	18
12	21H51A0525	THOTA MAHESHWARI	5	13	18
13	21H51A0526	VEERELLI SAIVENKATA REDDY	AB	AB	AB
14	21H51A0529	BELKONI ANVESH	5	11	16
15	21H51A0533	DASARI AJAY KUMAR	AB	14	14
16	21H51A0537	GANTA NISHAL	5	11	16
17	21H51A0540	KOMMANABOINA ANUSHA	5	08	13
18	21H51A0541	LOKOTI SRICHARAN	5	10	15
19	21H51A0542	M KAVYA	5	12	17
20	21H51A0544	OJAS RAKESH GARPALLIWAR	AB	10	10
21	21H51A0545	PEDDINTI SAI VARDHAN	5	09	14
22	21H51A0547	SATVIKA KARUMUDI	5	14	19
23	21H51A0549	THAMMISHETTY SHASHANK	5	07	12
24	21H51A0550	TUMMALA VENGAL RAYUDU	5	11	16
25	21H51A0551	UMMEDA SHIVA SAI KRISHNA	5	07	12
26	21H51A0552	VEMULA PRIYA PRAMIDHA	5	12	17
27	21H51A0554	ABHISHEK KUMAR SINGH	5	15	20
28	21H51A0555	ALETI ASHWITHA REDDY	5	12	17
29	21H51A0556	BATTU VICTOR DINAKAR BABU	5	08	13
30	21H51A0559	GANDRATH SRI YAGNA	AB	08	08
31	21H51A0562	JOGU TARUN TEJA	AB	07	07
32	21H51A0563	KARRA VINAY REDDY	5	03	08
33	21H51A0569	MOHAMMAD FERIA	5	09	14
34	21H51A0570	NAGULAPALLY UDAYKIRAN	5	08	13
35	21H51A0572	SARVADEY ZANETA	5	06	11

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
36	21H51A0573	SATHYARAM DHANA LAKSHMI	5	12	17
37	21H51A0574	SHA SOPNIL JAIN	5	10	15
38	21H51A0578	VUPPALA SHLAGHA	5	12	17
39	21H51A0582	JYOTHI BALAJI	5	06	11
40	21H51A0583	K RITIKA REDDY	5	17	22
41	21H51A0584	KOPPULA VENKATA SAI NANDINI	5	06	11
42	21H51A0586	M GANESH	5	11	16
43	21H51A0592	NENAVATH SRAVANI RATHOD	5	10	15
44	21H51A0595	PAVAN KUMAR	AB	~~AB~~ 18	~~AB~~ 18 Nav.
45	21H51A0597	ROSHAN TALARI	5	14	19
46	21H51A0598	S VARUN	5	09	14
47	21H51A05A5	AILENI SATHWIK	AB	05	05
48	21H51A05A6	AKURATHI RITHVIK SESHAGIRI	5	07	12
49	21H51A05A9	BIJJAM SOUMIKA	5	02	07
50	21H51A05B0	BODA ASHOK	AB	05	05
51	21H51A05B6	GOLLAPUDI NITHIN	5	05	10
52	21H51A05C1	NALLAKULA KIRANKUMAR	5	04	09
53	21H51A05C4	RITVIK PRATHAPANI	5	05	10
54	22H55A0501	AILLURI AMARDEEP REDDY	5	12	17
55	22H55A0502	BAIROJU SINDHU	5	14	19
56	22H55A0503	BODA AVINASH	5	04	09
57	22H55A0504	BODA RAHUL SAI KIRAN	AB	04	04
58	22H55A0505	CHAKILAM BHARAT KUMAR	5	12	17
59	22H55A0506	ERLA VENU	5	07	12
60	22H55A0507	JONNALA SOWMYA	5	11	16
61	22H55A0508	KALE PRABHAS	5	07	12
62	22H55A0509	KATKAM MANASWINI	5	10	15
63	22H55A0510	KODIDALA KOMALI	5	06	11
64	22H55A0511	KONDA MAHIMASRI	5	08	13
65	22H55A0512	KONDAPARTHI MANJEERA	5	15	20
66	22H55A0513	KUMMARI RAJESH	5	13	18
67	22H55A0514	KURUMULA LOKESH	5	08	13

Name&Signature of the Faculty :	P. Navya Sree
Department :	CSE
Mobile No :	8555059731

HOD/CSE

CMR College of Engineering & Technology

(UGC AUTONOMOUS)

Kandlakoya , Medchal Road - 501401

Department of Computer Science and Engineering

MID-II MARKS LIST

Class : III B.Tech. I SEM CSE SECTION-B A.Y.2023-24

SUBJECT : Artificial Intelligence

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
1	21H51A0502	DASARI HARINI	5	14	19
2	21H51A0504	GAJULAPALLE SREE LAKSHMI	5	19	24
3	21H51A0506	J AKANSH	5	~~AB~~ 22	~~05~~ 27 Nag
4	21H51A0507	K ZAYD AHMED	5	11	16
5	21H51A0509	KURAPATI ESHWAR	5	14	19
6	21H51A0510	LAVANGU VAISHNAVI	5	16	21
7	21H51A0511	MAHANTHI SAI MANYA SRI	5	17	22
8	21H51A0512	MANAS CHHATWAL	5	15	20
9	21H51A0513	MANGINA SRI VENKATA SAI	5	12	17
10	21H51A0516	NAGIREDDY ANVITHA	5	18	23
11	21H51A0517	PADALA ANIL KUMAR	5	10	15
12	21H51A0522	SHREYASH SANJEEV KUMAR	5	12	17
13	21H51A0523	SIDDAMSHETTI SUMITH	5	9	14
14	21H51A0527	AKSHAT KALA	5	13	18
15	21H51A0528	ALAVALA KAVYA	5	18	23
16	21H51A0530	BENKI JYOTHIKA	5	17	22
17	21H51A0531	BENKI VARSHITHA RANI	5	22	27
18	21H51A0532	BOLLU HARI CHARHAN	5	10	15
19	21H51A0534	DAVULURI SAI SUJAN	5	16	21
20	21H51A0535	DESHAPATHI SAHITHI	5	15	20
21	21H51A0536	DHULIPALLA VENKATA SAI SIVA	5	14	19
22	21H51A0539	KOLAN SAHASRA REDDY	5	12	17
23	21H51A0543	MANGA TARAKA RATNA YOSHITH	5	16	21
24	21H51A0546	SAPNA TIWARI	5	16	21
25	21H51A0548	THAKUR ABHINAV SINGH	5	15	20
26	21H51A0553	ABBULA VINUTHNA	5	16	21
27	21H51A0557	BUCHENELLI NIKHILESH REDDY	5	14	19
28	21H51A0558	DANDA VENKATA SATHWIK REDDY	5	13	18
29	21H51A0560	GORINTA RAHULU	5	14	19
30	21H51A0561	GUNREDDY AKSHITH REDDY	5	12	17
31	21H51A0564	KODURU PRANATHI	5	19	24
32	21H51A0565	KONDA VISHAL GOUD	5	12	17
33	21H51A0566	KURAKULA SHAILESH	5	15	20
34	21H51A0567	MADIRA SAI RISHITHA	5	19	24
35	21H51A0568	MANURI CHANDU BABU	5	12	17

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
36	21H51A0571	NIMMALA SAI	5	12	17
37	21H51A0575	TUDURU SATHWIK	5	13	18
38	21H51A0576	U NAGA MANASWINI	5	14	19
39	21H51A0577	VARLA RAMAKRISHNA REDDY	5	15	20
40	21H51A0579	AMBATI ROHITH RAJU	5	17	22
41	21H51A0580	BAIRA ANUSHA	5	18	23
42	21H51A0581	GUNNALA AKHILA	5	21	26
43	21H51A0585	KUDUMULA ANVESH REDDY	5	14	19
44	21H51A0587	MANDALOJU VASANTH KUMAR	5	14	19
45	21H51A0588	MOHAMMAD ABDUL KALAM	5	10	15
46	21H51A0589	MOHAMMED MUDASSIR ALI	5	10	15
47	21H51A0590	NALABOLU MOUNIKA	5	13	18
48	21H51A0591	NAMPALLY SIDDHARTHA	5	14	19
49	21H51A0593	PAMULA BEULAH SUPRAGNYA	AB	14	14
50	21H51A0594	PANCHAGNULA VINUTNA	5	16	21
51	21H51A0596	RAGE DAMODHAR	5	13	18
52	21H51A0599	SAI KIRAN B L S ·	5	16	21
53	21H51A05A0	SHESHAVAMATAM SUCHIT PAUL	5	13	18
54	21H51A05A1	TEEGALA BHANU TEJA REDDY	5	15	20
55	21H51A05A2	VADDI RISHIKA	5	23	28
56	21H51A05A3	YADDANAPUDI VISHNU SRIVATSAVA	5	16	21
57	21H51A05A4	YELDI ARUN	5	15	20
58	21H51A05A7	BAJRANG HARSH SINGH	5	16	21
59	21H51A05A8	BASAR SHYAM SUNDER RAO	5	14	19
60	21H51A05B1	BUNNI SHARANYA	5	17	22
61	21H51A05B2	C J VISHNU PRAKASH	5	14	19
62	21H51A05B3	CHIMMULA SHIVA PRASAD REDDY	5	12	17
63	21H51A05B4	DOLLA RENUKA	5	14	19
64	21H51A05B5	ERUKULA RAJASREE	5	13	18
65	21H51A05B7	HARIKA REDDY GANTA	5	10	15
66	21H51A05B8	INDUPALLI SHARONSUDHA	5	14	19
67	21H51A05B9	MADULAPURAM SAI YASHWANTH RAJ	5	16	21
68	21H51A05C0	MALLELA SINDHUJA	5	18	23
69	21H51A05C2	RANGU ABHINAV	5	16	21
70	21H51A05C3	RAYABARAPU CHATHURYA	5	14	19
71	21H51A05C5	SEGU JAYA BALA HARSHAVARDHAN	5	15	20
72	21H51A05C8	THATIKONDA AKHILA	5	17	22
73	21H51A05C9	VAKALA KAVYA SAI SUMA SRI	5	15	20
74	21H51A05D1	ANUJ KUMAR	5	15	20
75	21H51A05D2	BACHAWAR VINITHA	5	17	22

Name&Signature of the Faculty : P. Navya Sree

Department : CSE

Mobile No : 8555059731

HOD/CSE

CMR College of Engineering & Technology

(UGC AUTONOMOUS)

Kandlakoya , Medchal Road - 501401

Department of Computer Science and Engineering

MID-II MARKS LIST

Class : III B.Tech. I SEM CSE SECTION-C A.Y.2023-24

SUBJECT : AI

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
1	21H51A05C6	SOMU KOTESWARA REDDY	5	17	22
2	21H51A05C7	SUNKAPAKA JOHN	5	14	19
3	21H51A05D0	VALLAMKONDA POOJITHA	5	20	25
4	21H51A05D3	BASHAM RAJU	5	19	24
5	21H51A05D4	BUSSA TEJASWINI	5	17	22
6	21H51A05D5	DADE DINISHA	5	21	26
7	21H51A05D6	DEEKONDA SAKETH	5	0	5
8	21H51A05E3	MANCHI AKSHAYA	5	22	27
9	21H51A05E4	MOHAMMAD ARSHAD NIZAMI	5	5	10
10	21H51A05F1	P Y GEETHA MADHURI	5	16	21
11	21H51A05F3	SHAIK ILLIYAZ	5	11	16
12	21H51A05F5	TUSHAR PUNIA	5	22	27
13	21H51A05F8	DODDI SAI PHANI HARI CHANDANA	5	17	23
14	21H51A05F9	GADUGULA KALYANI	5	18	23
15	21H51A05G2	IYLA SNEHARIKA	5	19	22
16	21H51A05G5	KANUGO NESHIT RAJ	5	12	17
17	21H51A05H2	PODDUTURI NITHIN REDDY	5	12	16
18	21H51A05H8	TADEM RAVITEJA	5	5	10
19	21H51A05J8	GUNTHAPALLI MALINI	5	12	17
20	21H51A05J9	GURRAM KRISHNA PRASANTH	5	Ab 21	26
21	21H51A05K3	KODIGANTI SAI KISHORE	0	2	2
22	21H51A05K8	SEELAMSETTY PRASANNA GAYATHRI	5	11	16
23	21H51A05L1	SRIRAM NAGARAJU	5	18	23
24	21H51A05L7	YALLA TEJASWIK REDDY	5	10	15
25	21H51A05L8	BEHARA SURAJ	5	14	19
26	21H51A05M0	CHILUKA SAI KARTHIK	5	20	25
27	21H51A05M1	DAMARLA HEMAVATHI	5	13	18
28	21H51A05M4	GIRAVENA ARYA	5	14	19
29	21H51A05M9	MOKIRALA JHANSI	5	17	22
30	21H51A05N1	NEELA SAI ADITYA	5	11	16
31	21H51A05N3	POTRU SAI NITISH	5	14	19
32	21H51A05N4	PRAHARSHITHA SURAGONI	5	15	20
33	21H51A05N5	PULI PRANEETH GOUD	5	13	17+1
34	21H51A05P0	TALOORI PRABHU KIRAN	5	16	21
35	21H51A05P2	VAVILLA RAVITEJA	5	22	27

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
36	21H51A05P4	ALLURI SAI SATHWIK REDDY	5	17	22
37	21H51A05P5	ANDE AJAY	5	15	20
38	21H51A05P7	BESTHA NANDA KISHORE	5	20	25
39	21H51A05P8	CHAVATAPALLI MUKUNDA SRI HASINI	5	12	17
40	21H51A05P9	CHEPYALA SATHWIK REDDY	5	11	16
41	21H51A05Q1	DAGGULA PRASHANTH	5	13	18
42	21H51A05Q2	GAJULA NAVANEETH	5	16	21
43	21H51A05Q3	GUDAPATI NITHIN KUMAR	5	11	16
44	21H51A05R3	PINAPATI ABHISHEK	5	11	16
45	21H51A05R4	RACHAMALLA SAI UJITHA REDDY	5	11	16
46	21H51A05R5	SATTU RAKESH	5	11	16
47	21H51A05R6	SHREYA M	5	11	16
48	21H51A05R7	YERAVELLI RUCHITHA	5	15	20
49	22H55A0515	M. SAI RANJITH REDDY	5	11	16
50	22H55A0516	MAHATHI DESAI	5	15	20
51	22H55A0517	MD TOWHEED	5	13	18
52	22H55A0518	MOHAMMED HANEF	5	15	20
53	22H55A0519	NAGARAM SHIVA CHAND	5	~~AB~~ 16	~~5~~ 21
54	22H55A0520	NARGE CHARANETEJA	5	17	22
55	22H55A0521	NEELAM RAMYA SARI	5	18	23
56	22H55A0522	PANDAV SONIA	5	18	23
57	22H55A0523	PATHLAVATH SUNITHA	5	19	24
58	22H55A0524	POTTIPALLY DEEPIKA	5	8	13
59	22H55A0525	PULIGANTI MAHENDAR	5	18	23
60	22H55A0526	SARDESHI PRAVEEN KUMAR	5	18	23
61	22H55A0527	VISLAVATH ANITHA	5	16	21

Name&Signature of the Faculty :	Vijayalaxmi. K
Department :	CSE (C)
Mobile No :	953593451

HOD/CSE

CMR College of Engineering & Technology

(UGC AUTONOMOUS)

Kandlakoya, Medchal Road - 501401

Department of Computer Science and Engineering

MID-II MARKS LIST

Class : III B.Tech. I SEM CSE SECTION-D A.Y.2023-24

SUBJECT : AI

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
1	21H51A05D7	DHUDURI SATHVIKA	5	21	26
2	21H51A05D8	GAMPALA SRI DURGA PRABHATH	5	14	19
3	21H51A05D9	GUNDLA VAMSHIDHAR	5	18	23
4	21H51A05E0	KASANAGOTTU AMULYA	5	19	24
5	21H51A05E1	KOTHA VAISHNAVI	5	17	22
6	21H51A05E2	KUMBALA ABHILASH REDDY	5	16	21
7	21H51A05E6	NAKKALA KEERTHANA	5	17	22
8	21H51A05E7	NEELAM BHARATH KUMAR	5	15	20
9	21H51A05E8	NEERUDI HARIPRASAD	5	21	26
10	21H51A05E9	ODURI VEERAMANIKANTA	5	15	20
11	21H51A05F0	OM GUPTA	5	17	22
12	21H51A05F2	ROHAN SACHIN RAKHE	5	17	22
13	21H51A05F4	SHAIK TASNIM	5	20	25
14	21H51A05F6	YARRAMSETTI MADHU VENKATA	5	20	25
15	21H51A05F7	BABBI THAPA	5	15	20
16	21H51A05G0	GUDIPALLY SAI SANJAY	5	15	20
17	21H51A05G1	GUNNA VINAY KUMAR REDDY	5	22	27
18	21H51A05G3	K SRI HARINI	5	22	27
19	21H51A05G4	KANDI SWETHA	5	22	27
20	21H51A05G6	KHANDESH THANU SRI	5	18	18+5
21	21H51A05G7	MAMIDI VENU GOPAL	5	22	27
22	21H51A05G8	MARAGONI KARTHIKEYA	5	13	18
23	21H51A05G9	NALIMELA JITHIN REDDY	5	14	19
24	21H51A05H1	PATRAYADI RAVI	5	15	20
25	21H51A05H3	POTHARAJU SAI KIRAN	5	20	25
26	21H51A05H5	SHERIKAR RAHUL	5	19	24
27	21H51A05H6	SOMARAJUPALLI THEJASWI	5	22	27
28	21H51A05H7	SUDAM SHIVA	5	20	25
29	21H51A05H9	THALLAM GEETHAN	5	20	25
30	21H51A05J0	TODUPUNURI SHAI PRIYA	5	22	27
31	21H51A05J1	TUMMALA SAHITH	5	12	17
32	21H51A05J2	VIJAYAGIRI AMULYA	5	22	27
33	21H51A05J3	ABHAY PRATAP SINGH	5	22	27
34	21H51A05J4	AYEMON ZEBA	5	17	22
35	21H51A05J5	BONDALA SRINATH	5	21	26

S.No	Roll Number	Name of the Candidate	Assignment (5M)	MID Marks (25 M)	Total (30 M)
36	21H51A05J6	DODDAPANENI MEGHAN CHOWDARY	5	21	26
37	21H51A05J7	GORANTI SANTHU SATHWIK	5	22	27
38	21H51A05K0	KACHIREDDY JAYASREE	5	22	27
39	21H51A05K1	KAJA SANJEEV KUMAR	5	24	29
40	21H51A05K2	KANTU ANANTHKUMAR	5	16	21
41	21H51A05K4	KONDETI VIKRAMREDDY	5	22	27
42	21H51A05K5	KRITIKA KHATRI	5	22	27
43	21H51A05K6	NITYANANDAYYA MATHPATHI	5	23	28
44	21H51A05K9	SHANIGALA VISHNU	5	~~29~~ 21	26
45	21H51A05L0	SINDEY ABHIGNA	5	21	26
46	21H51A05L2	SUMESH	5	20	25
47	21H51A05L3	TANNIRU MAHESH	5	20	25
48	21H51A05L4	TUSYAA SREERALA	5	19	24
49	21H51A05L5	UNI SAILESH	5	23	28
50	21H51A05L6	VAGUAMRI SRINANDHAN	5	23	28
51	21H51A05L9	BHAKE SHASHANK	5	22	27
52	21H51A05M2	DIVYA GAUTAM	5	21	26
53	21H51A05M3	GANGASANI SHANKARSHAN	5	23	28
54	21H51A05M5	GUMMIREDDY SAINATH REDDY	5	21	26
55	21H51A05M6	KALLURI THANMAI	5	16+1	22
56	21H51A05M7	KATRAVATH MANJULA	5	22	27
57	21H51A05M8	MOHAMMED SAMEER ALI	5	19	24
58	21H51A05N0	NANCHARLA SAI AKSHITHA	5	23	28
59	21H51A05N2	OLIGE RANI	5	22	~~25~~ 27
60	21H51A05N6	SAKKERLA RAJ KUMAR	5	13	18
61	21H51A05N7	SALENDRA MANOJ KUMAR	5	18	23
62	21H51A05N8	SHAIK JAVED	5	21	26
63	21H51A05N9	SHRIYA MALANI	5	23	28
64	21H51A05P1	VASURI VINAY KUMAR	5	20	25
65	21H51A05P3	VITTAPUR BINNU REDDY	5	23	28
66	21H51A05P6	BANOTHU DALI HIMASRI	5	~~Ab~~ 17	22
67	21H51A05Q0	D GAYATHRI	5	23	28
68	21H51A05Q4	GUDIPUDI DHEERAJ	5	19	24
69	21H51A05Q5	GURRAM SRIKANTH	5	19	24
70	21H51A05Q6	KALVAKUNTA CHANDRASHEKAR	5	17	22
71	21H51A05Q7	KAPU HARSHA VARDAN REDDY	5	24	29
72	21H51A05Q8	KOTTE MOUNIKA	5	24	29
73	21H51A05Q9	MANDA VIGHNESHWARA REDDY	5	19	24
74	21H51A05R0	MANDHUMULA DEEPAK	5	20+1	26
75	21H51A05R1	PEDDI PRAVALIKA REDDY	5	~~Ab~~ 20	25
76	21H51A05R2	PENDEM YOGITHA	5	15	20
77	21H51A05R8	YESUGARI ADHARSH	5	18	23

Name&Signature of the Faculty : Vijayalaxmi. K

Designation: Asst. Proff

Department : CSE (D)

Mobile No : 9535934513

HOD/CSE

SET-2

Hall Ticket No.

Question Paper Code: A30530

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

CMR

B.Tech V Semester Mid-I Examinations November -2023

(Regulation: CMRCET-R18)

Subject Name: Artificial Intelligence | Time: 1.30 PM to 3.10 PM

Date: 03-11-2023 | Branch: CSE | Max Marks: 25

PART A

Answer all FIVE questions (Compulsory)

Each question carries TWO marks. 5x2=10M

Question Number	Question Format	CO	B.T Level
1	What is Artificial Intelligence and Natural Intelligence?	CO1	BT3
2	Describe Heuristic & Heuristic function.	CO1	BT3
3	State the Knowledge Base?	CO2	BT1
4	Interpret the concept of entailment	CO2	BT4
5	Define first order logic?	CO3	BT1

PART B

Answer ALL questions.

Each question carries FIVE Marks.

3x5=15M

Question Number	Question Format	CO	B.T Level
6A	Discuss Depth First Search strategy and Iterative Deepening with suitable example ?	CO1	BT3
	OR		
6B	What is an Agent? Explain the types of agents briefly?	CO1	BT2
7A	Investigate Resolution rule. Prove by resolution $((P \vee Q) \cap (P \rightarrow R) \cap (Q \rightarrow R)) \supset R$	CO2	BT6
	OR		
7B	Define Constraint Satisfaction Problem using Backtracking Algorithm with example?	CO2	BT1
8A	Discuss Simulated Annealing with an example?	CO3	BT2
	OR		
8B	Define Alpha-Beta pruning with an example?	CO3	BT1

MID -I

1) What is artificial intelligence and Natural Intelligence?

Artificial intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence.

Natural intelligence is the behavior of the human being that has come from past experiences and the doings based upon situation, and environment. And it is completely based upon the ability to change his/her surroundings through knowledge which we gained.

2) Describe Heuristic and Heuristic function.

A heuristic is a technique that is used to solve a problem faster than the classic methods. These techniques are used to find the approximate solution of a problem when classical methods do not. Heuristics are said to be the problem-solving techniques that result in practical and quick solutions.

A heuristic function in artificial intelligence, is an evaluation function used to estimate the cost or potential of reaching a goal state from a given state in a problem-solving domain.

3) State the knowledge base.

An AI knowledge base is a centralized repository for information integrated with artificial intelligence (AI) technologies. Unlike traditional knowledge bases, which serve as static repositories of information like FAQs, articles, and how-to guides, an AI knowledge base is dynamic. It leverages machine learning and natural language processing to learn from various interactions such as website behavior and customer feedback, enhancing its capability to deliver precise and beneficial information over time.

Knowledge-based systems typically have three components, which include:

- **Knowledge base:** A knowledge base is an established collection of information and resources. The system uses this as its repository for the knowledge it uses to make decisions.

4) Interpret the concept of entailment.

$\alpha \models \beta$ means α entails β i.e. β follows logically from α, where α and β are sentences Mathematically, $\alpha \models \beta$ if and only if in every model in which α is true, β is also true. Another way: if α is true, then β must also be true.

5) Define First order Logic.

First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.First-order logic is also known as **Predicate logic or First-order predicate logic**. First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.

a. **Objects:** A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, etc

b. **Relations: It can be unary relation such as:** red, round, is adjacent, **or n-any relation such as:** the sister of, brother of, has color, comes between

c. **Function:** Father of, best friend, third inning of, end of, ..

6A) Discuss DFS strategy and iterative deepening with suitable examples.

Depth First Search for a graph is an important part of the uninformed search strategy. This algorithm is used to perform search activities without any domain knowledge. The algorithm works so that it searches the nodes in a depth-wise fashion.

The step by step process to implement the DFS traversal is given as follows -

1. First, create a stack with the total number of vertices in the graph.
2. Now, choose any vertex as the starting point of traversal, and push that vertex into the stack.
3. After that, push a non-visited vertex (adjacent to the vertex on the top of the stack) to the top of the stack.
4. Now, repeat steps 3 and 4 until no vertices are left to visit from the vertex on the stack's top.
5. If no vertex is left, go back and pop a vertex from the stack.

6. Repeat steps 2, 3, and 4 until the stack is empty.

Algorithm

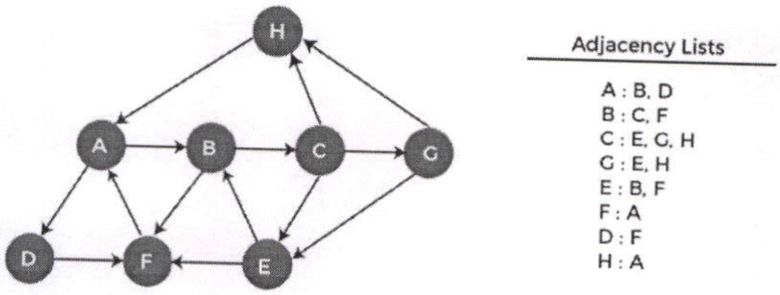
Step 1: SET STATUS = 1 (ready state) for each node in G

Step 2: Push the starting node A on the stack and set its STATUS = 2 (waiting state)

Step 3: Repeat Steps 4 and 5 until STACK is empty

Step 4: Pop the top node N. Process it and set its STATUS = 3 (processed state)

Step 5: Push on the stack all the neighbors of N that are in the ready state (whose STATUS = 1) and set their STATUS = 2 (waiting state)



Now, let's start examining the graph starting from Node H.

Step 1 - First, push H onto the stack.

1. STACK: H

Step 2 - POP the top element from the stack, i.e., H, and print it. Now, PUSH all the neighbors of H onto the stack that are in ready state.

1. Print: H]STACK: A

Step 3 - POP the top element from the stack, i.e., A, and print it. Now, PUSH all the neighbors of A onto the stack that are in ready state.

1. Print: A
2. STACK: B, D

Step 4 - POP the top element from the POP the top element from the stack, i.e., D, and print it. Now, PUSH all the neighbors of D onto the stack that are in ready state.

1. Print: D
2. STACK: B, F

Step 5 - POP the top element from the stack, i.e., F, and print it. Now, PUSH all the neighbors of F onto the stack that are in ready state.

1. Print: F
2. STACK: B

Step 6 - POP the top element from the stack, i.e., B, and print it. Now, PUSH all the neighbors of B onto the stack that are in ready state.

1. Print: B
2. STACK: C

Step 7 - POP the top element from the stack, i.e., C, and print it. Now, PUSH all the neighbors of C onto the stack that are in ready state.

1. Print: C
2. STACK: E, G

tep 8 - POP the top element from the stack, i.e., G and PUSH all the neighbors of G onto the stack that are in ready state.

1. Print: G
2. STACK: E

Step 9 - POP the top element from the stack, i.e., E and PUSH all the neighbors of E onto the stack that are in ready state.

1. Print: E
2. STACK:

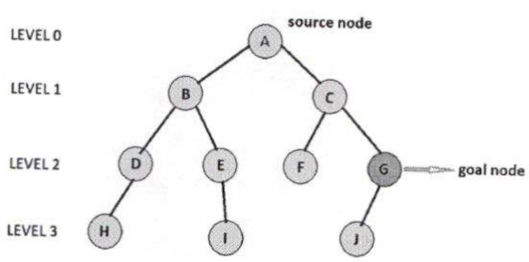
Now, all the graph nodes have been traversed, and the stack is empty.

The time complexity of the DFS algorithm is **O(V+E)**, where V is the number of vertices and E is the number of edges in the graph.

IDDFS combines depth-first search's space-efficiency and breadth-first search's fast search (for nodes closer to root).

How does IDDFS work?

IDDFS calls DFS for different depths starting from an initial value. In every call, DFS is restricted from going beyond given depth. So basically we do DFS in a BFS fashion



IDDFS with max depth-limit = 3
Note that iteration terminates at depth-limit=2
Iteration 0: A
Iteration 1: A->B->C
Iteration 2: A->B->D->E->C->F->G

6 B) What is an agent? Explain types of agents briefly.

An agent is anything that can be viewed as **perceiving its environment through sensors** and acting upon that environment through **actuators**.

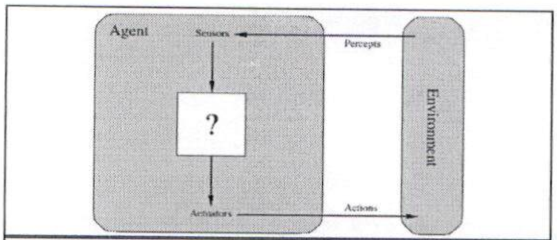
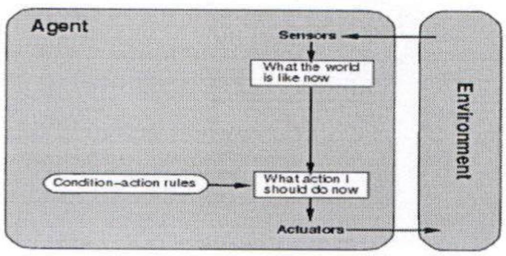


Figure 2.1 Agents interact with environments through sensors and actuators.

Types of AI Agents :

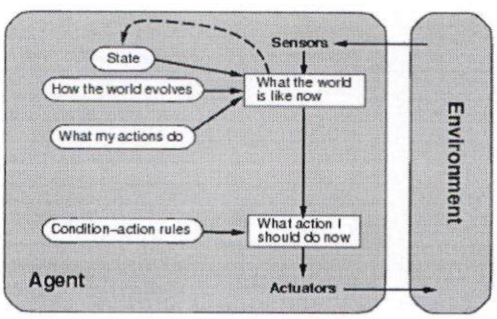
i) Simple Reflex Agents :

- Acts only on the basis of current percept.
- Does not consider any part of precepts history during their decision and action process.
- It works on condition – action rule.



ii)Model based reflex agent :

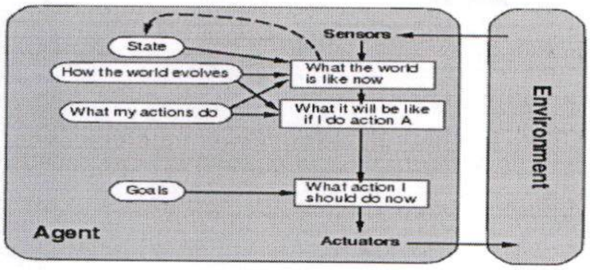
- The Model-based agent can work in a **partially observable environment**, and track the situation.



iii) Goal-based agent

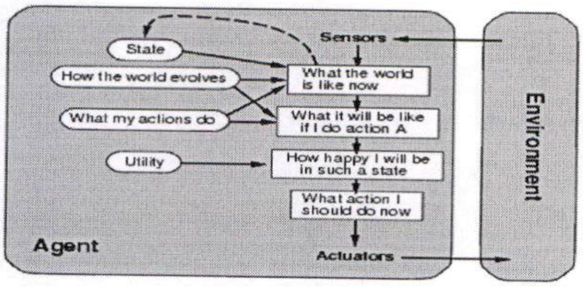
- Goal-based AI agents are an expansion of model-based AI agents. The whole technique of the goal-based agent to reach a goal or a final state it is based on searching and planning.

- Agent selects its actions in order to achieve a goal.
- Handles partially observable environments.



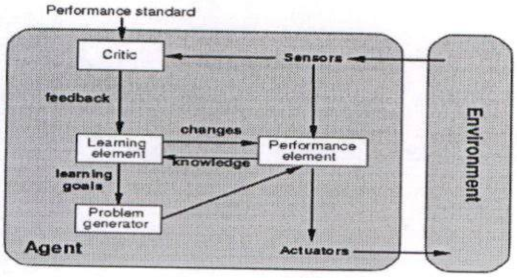
iv) Utility based Agents

- When there are multiple possible alternatives, then to decide which one is best, utility –based agents are used.
- They choose actions based on a preference(utility) for each state.
- Utility measures how happy or unhappy the agent is.
- It handles partially observable environments.



v) Learning Agent :

- Agents can learn from past experiences or it has learning capabilities.
- Starts with basic knowledge and then it is able to act and adopt automatically through learning.
- Learning agent perceives the environment and keeps track of every information.



7A) Investigate resolution rule.

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. It was invented by Mathematician John Alan Robinson in 1965.

Resolution is used, if various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the conjunctive normal form or clausal form.

Clause: Disjunction of literals (an atomic sentence) is called a clause. It is also known as a unit clause.

Conjunctive Normal Form: A sentence represented as a conjunction of clauses is said to be conjunctive normal form or CNF.The resolution rule for first-order logic is simply a lifted version of the propositional rule. Resolution can resolve two clauses if they contain complementary literals, which are assumed to be standardized apart so that they share no variables.

This rule is also called the binary resolution rule because it only resolves exactly two literals.

Steps for Resolution:

1. Conversion of facts into first-order logic
2. Convert FOL statements into CNF
3. Negate the statement which needs to prove (proof by contradiction)
4. Draw resolution graph (unification)

7B)Define constraint satisfaction problems using backtracking.

Finding a solution that meets a set of constraints is the goal of constraint satisfaction problems (CSPs). For tasks including resource allocation, planning, scheduling, and decision-making, CSPs are frequently employed in AI.

There are mainly three basic components in the constraint satisfaction problem:

Variables: The things that need to be determined are variables. Variables in a CSP are the objects that must have values assigned to them in order to satisfy a particular set of constraints.

Domains: The range of potential values that a variable can have is represented by domains.

Constraints: The guidelines that control how variables relate to one another are known as constraints.

```
function BACKTRACKING-SEARCH(csp) returns a solution, or failure
  return BACKTRACK({ }, csp)

function BACKTRACK(assignment, csp) returns a solution, or failure
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(csp)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
     if value is consistent with assignment then
        add {var = value} to assignment
        inferences ← INFERENCE(csp, var, value)
        if inferences ≠ failure then
           add inferences to assignment
           result ← BACKTRACK(assignment, csp)
           if result ≠ failure then
              return result
     remove {var = value} and inferences from assignment
  return failure
```

- **Figure 6.5** A simple backtracking algorithm for constraint satisfaction problems. The algorithm is modeled on the recursive depth-first search of Chapter 3. By varying the functions SELECT-UNASSIGNED-VARIABLE and ORDER-DOMAIN-VALUES, we can implement the general-purpose heuristics discussed in the text. The function INFERENCE can optionally be used to impose arc-, path-, or k-consistency, as desired. If a value choice leads to failure (noticed either by INFERENCE or by BACKTRACK), then value assignments (including those made by INFERENCE) are removed from the current assignment and a new value is tried.

8A) Discuss Simulated Annealing with an example.

Simulated annealing is a method for solving complex problems in artificial intelligence. It is a guided search procedure that allows the computer to gradually explore different solutions to a problem until it finds one that works best.

In artificial intelligence, simulated annealing is a heuristic search method that is often used to find a good solution to a problem. The algorithm works by repeatedly altering the solution until it improves upon the current best solution found by the computer. This process is repeated until the improvement slows or stops.

The advantages of simulated annealing in artificial intelligence

Simulated annealing is a relatively new algorithm used in artificial intelligence that aims to solve problems by alternating between different solutions until a satisfactory solution is found. The advantages of using simulated annealing in artificial intelligence are that it is relatively fast, easy to learn, and can produce good results even for difficult problems. Additionally, simulated annealing is versatile and can be used for a variety of problems, making it a good tool for AI development.

8B) Define alpha beta pruning with an example.

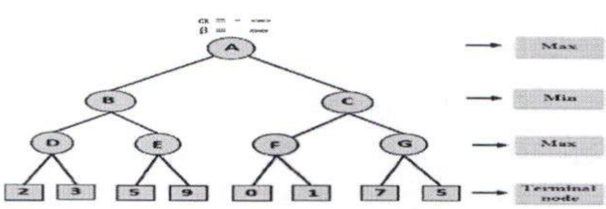
Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.As we have seen in the minimax search algorithm, the number of game states it has to examine are exponential in depth of the tree. Since we cannot eliminate the exponent, but we can cut it to half. Hence there is a technique by which without checking each node of the game tree we can compute the correct minimax decision, and this technique is called **pruning**. This involves two threshold parameters Alpha and Beta for future expansion, so it is called **alpha-beta pruning**.

a. **Alpha:** The best (highest-value) choice we have found so far at any point along the path of Maximizer. The initial value of alpha is $-\infty$.

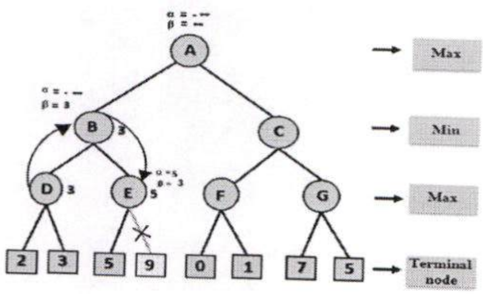
b. **Beta:** The best (lowest-value) choice we have found so far at any point along the path of Minimizer. The initial value of beta is $+\infty$.

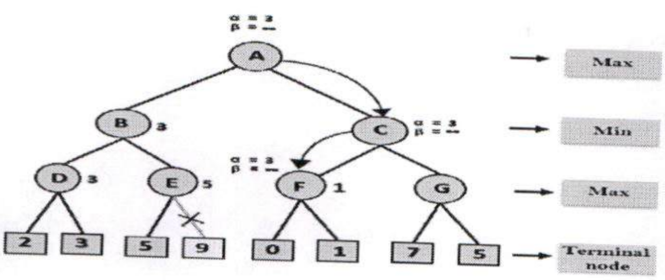
he main condition which required for alpha-beta pruning is:

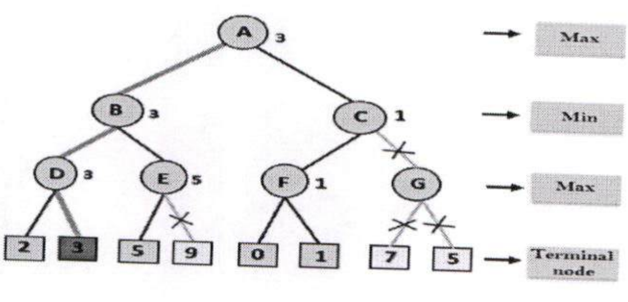
$\alpha >= \beta$



At Node D, the value of α will be calculated as its turn for Max. The value of α is compared with firstly 2 and then 3, and the max (2, 3) = 3 will be the value of α at node D and node value will also 3.Now algorithm backtrack to node B, where the value of β will change as this is a turn of Min, Now $\beta = +\infty$, will compare with the available subsequent nodes value, i.e. min (∞, 3) = 3, hence at node B now $\alpha = -\infty$, and $\beta = 3$


A
B 3
C
D 3
E 5
F
G
2
3
5
9
0
1
7
5
Max
Min
Max
Terminal node


A
B 3
C
D 3
E 5
F 1
G
2
3
5
9
0
1
7
5
Max
Min
Max
Terminal node


A 3
B 3
C 1
D 3
E 5
F 1
G
2
3
5
9
0
1
7
5
Max
Min
Max
Terminal node

Hall Ticket No. | | | | | | | | | | |

Question Paper Code: A30530

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

B.Tech V Semester Mid-II Examinations December -2023

(Regulation: CMRCET-R18)

Subject Name: Artificial Intelligence | Time: 1.30 PM to 3.10 PM

Date: 30-12-2023 | Branch: CSE | Max Marks: 25

PART A

Answer all FIVE questions (Compulsory)

Each question carries TWO marks. 5x2=10M

1 Explain about Ontological Engineering.

2 Define classical planning with block world problem.

3 Summarize Contingent planning with an example.

4 What is uncertainty and explain axioms of probability

5 Summarize supervised, unsupervised and reinforcement learning.

PART B

Answer ALL questions.

Each question carries FIVE Marks. 3x5=15M

6A Explain resolution in FOL with an example.

OR

6B What is Reasoning with Default Information?

7A Summarize Hierarchical planning approach with example.

OR

7B Explain different algorithms for Planning with State-Space Search with example.

8A Explain decision tree with the example wait for a table in restaurant.

OR

8B Explain Baye's rule and its uses with some examples.

END

Hall Ticket No. | | | | | | | | | | |

Question Paper Code: A30530 29

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

B.Tech V Semester Mid-II Examinations December -2023

(Regulation: CMRCET-R18)

Subject Name: Artificial Intelligence | Time: 1.30 PM to 3.10 PM

Date: 30-12-2023 | Branch: CSE | Max Marks: 25

PART A

Answer all FIVE questions (Compulsory)

Each question carries TWO marks. 5x2=10M

1 Explain about Ontological Engineering.

2 Define classical planning with block world problem.

3 Summarize Contingent planning with an example.

4 What is uncertainty and explain axioms of probability

5 Summarize supervised, unsupervised and reinforcement learning.

PART B

Answer ALL questions.

Each question carries FIVE Marks. 3x5=15M

6A Explain resolution in FOL with an example.

OR

6B What is Reasoning with Default Information?

7A Summarize Hierarchical planning approach with example.

OR

7B Explain different algorithms for Planning with State-Space Search with example.

8A Explain decision tree with the example wait for a table in restaurant.

OR

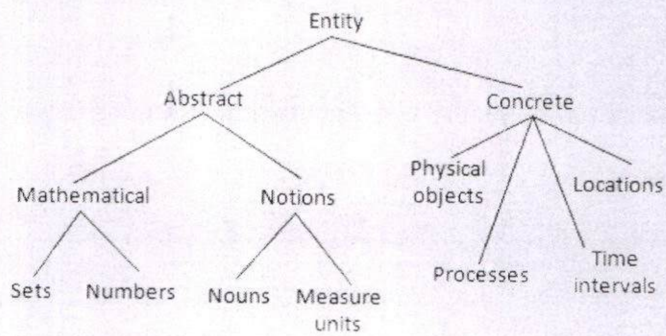
8B Explain Baye's rule and its uses with some examples.

END

MID -II

1) Explain about ontological engineering.

Ontologies are formal definitions of vocabularies that allow us to define difficult or complex structures and new relationships between vocabulary terms and members of classes that we define. Ontologies generally describe specific domains such as scientific research areas.



2Q)Define classical planning with a block world problem.

Classical planning is the problem of finding sequence of actions for achieving a goal from an initial state assuming that actions have deterministic effects.

$Init(On(A, Table) \wedge On(B, Table) \wedge On(C, A)$
$\wedge Block(A) \wedge Block(B) \wedge Block(C) \wedge Clear(B) \wedge Clear(C))$
$Goal(On(A, B) \wedge On(B, C))$
$Action(Move(b, x, y),$
PRECOND: $On(b, x) \wedge Clear(b) \wedge Clear(y) \wedge Block(b) \wedge Block(y) \wedge$
$(b \neq x) \wedge (b \neq y) \wedge (x \neq y),$
EFFECT: $On(b, y) \wedge Clear(x) \wedge \neg On(b, x) \wedge \neg Clear(y))$
$Action(MoveToTable(b, x),$
PRECOND: $On(b, x) \wedge Clear(b) \wedge Block(b) \wedge (b \neq x),$
EFFECT: $On(b, Table) \wedge Clear(x) \wedge \neg On(b, x))$

Figure 10.3 A planning problem in the blocks world: building a three-block tower. One solution is the sequence [*MoveToTable*(C, A), *Move*(B, *Table*, C), *Move*(A, *Table*, B)].

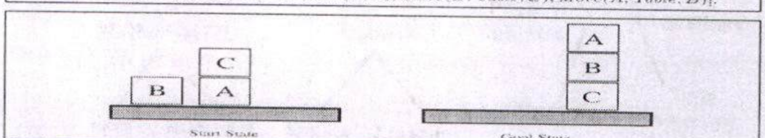


Figure 10.4 Diagram of the blocks-world problem in Figure 10.3.

3Q)Summarize contingent planning.

Conditional planning, also known as contingency planning, for partially observable and nondeterministic environments. Decides which action to choose based on special sensing actions that become parts of the plan.

4Q) What is uncertainty and explain axioms of probability.

The agent can never be completely certain about the state of the external world since there is ambiguity and uncertainty.

Reasons for uncertainty:

Sensors have limited precision, limited accuracy.

Partial information

Axioms of Probability:

1. $0 \leq P(a) \leq 1$: probabilities are between 0 and 1 inclusive
2. P(true) = 1, P(false) = 0
3. P (a U b) = P(a) + P(b) – P (a U b)

5Q)Summarize supervised,unsupervised and reinforcement learning.

Supervised learning :Supervised machine learning is used for making predictions from data. To be able to do that, we need to know what to predict, which is also known as the target variable.

Unsupervised learning:Imagine receiving swathes of data with no obvious pattern in it. A dataset with no labels or target values cannot come up with an answer to what to predict.

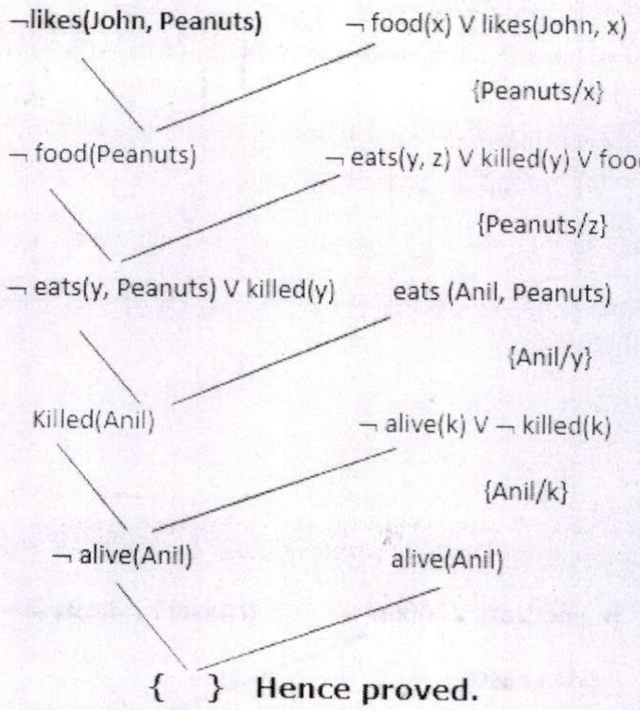
Reinforcement learning : In reinforcement learning, algorithms learn in an environment on their own. The field has gained quite some popularity over the years and has produced a variety of learning algorithms

6Q) Explain resolution in FOL with example

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proof by contradiction.

Steps for Resolution:

1. Conversion of facts into first-order logic.
2. Convert FOL statements into CNF
3. Negate the statement which needs to prove (proof by contradiction)
4. Draw resolution graph (unification).



6B) What is reasoning with default information?

The reasoning is the mental process of deriving logical conclusions and making predictions from available knowledge, facts, and beliefs. Or we can say, "Reasoning is a way to infer facts from existing data." It is a general process of thinking rationally, to find valid conclusions.

When giving information, we don't want to enumerate all of the exceptions, even if we could think of them all. In default reasoning, we specify general knowledge and modularly add exceptions. The general knowledge is used for cases we don't know are exceptional. Classical logic is monotonic: If g logically follows from A, it also follows from any superset of A. Default reasoning is nonmonotonic: When we add that something is exceptional, we can't conclude what we could before.

7A) Summarize hierarchical planning approach with example.

Hierarchical planning in artificial intelligence (AI) is a planning approach that involves organizing tasks and actions into multiple levels of abstraction or hierarchy, where higher-level tasks are decomposed into a sequence of lower-level tasks.

Components of Hierarchical Planning

- High-level goals: The overall objectives or tasks that the AI system aims to achieve.
- Task decomposition: Breaking down high-level goals into lower-level tasks or subgoals.
- Planning hierarchy: The organization of tasks or subgoals into a hierarchical structure, such as a tree or a directed acyclic graph (DAG).

- Plan generation at different levels: Reasoning and planning at different levels of the hierarchy, with plans generated for achieving subgoals or actions.
- Plan synthesis: Combining the plans for achieving subgoals or actions into a cohesive plan for execution.
- Plan execution: Carrying out the actions or subgoals in the plan in the correct order.

Advantages of Hierarchical Planning

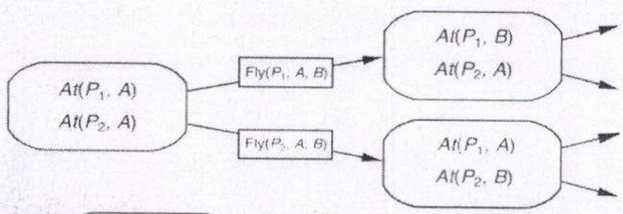
Flexibility
Scalability

7B) Explain different algorithms for planning with state space search using examples.
Planning state space consists of the initial state, set of goal states, set of actions , set of states and the path cost. This state space needs to be searched to find a sequence of actions leading to the goal state. This can be done in the forward or backward direction.

Forward State Space Search :

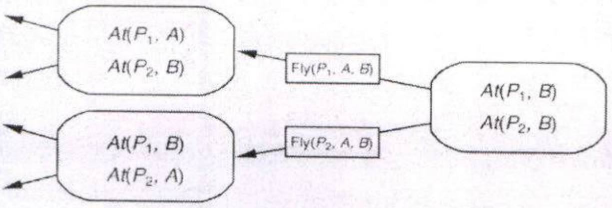
It is also called **Progression**. It starts from the initial state and searches in the forward direction till we reach the goal. It uses STRIPS representation.

- **Initial state**: Start state
- **Actions**: Each action has a particular **precondition** to be satisfied before the action can be performed and an **effect** that the action will have on the environment.
- **Goal test:** To check if the current state is the goal state or not.
- **Step cost:** Cost of each step which is assumed to be 1.



Backward State Space Search :

- It is also called Regression. It uses STRIPS representation. The problem formulation is similar to that of FSSS and consists of the initial state, actions, goal test and step cost. In BSSS, the searching starts from the goal state, and moves in the backward direction until the initial state is reached. It starts at the goal, checks if it is the initial state. If not, it applies the inverse of the actions to produce sub goals until the start state is reached. Below fig. shows backward search.



8A) Explain decision tree with example.

A decision tree is a type of supervised learning algorithm that is commonly used in machine learning to model and predict outcomes based on input data. It is a tree-like structure where each internal node tests on attribute, each branch corresponds to attribute value and each leaf node represents the final decision or prediction. The decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.

Alternate: whether there is a suitable alternative restaurant nearby.

Bar: whether the restaurant has a comfortable bar area to wait in.

Fri/Sat: true on Fridays and Saturdays.

Hungry: whether we are hungry.

Patrons: how many people are in the restaurant (values are None, Some, and Full)

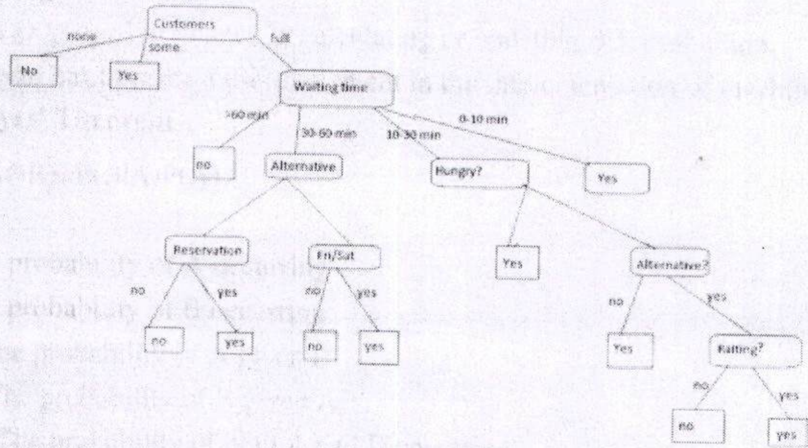
Price: the restaurant's price range

Raining: whether it is raining outside.

Type: the kind of restaurant (French, Italian, Thai, or burger).

Wait Estimate: the wait estimated by the host

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T



8B) Explain bayes rule and its uses with example.

- Bayes' Theorem allows you to update the predicted probabilities of an event by incorporating new information.
- It often is employed in finance in calculating or updating risk evaluation.
- The theorem has become a useful element in the implementation of machine learning.

Formula for Bayes' Theorem

P(A|B)P(B)=P(A∩B)=P(B|A)P(A).

where:

$P(A)$= The probability of A occurring

$P(B)$= The probability of B occurring

$P(A \mid B)$=The probability of A given B

$P(B \mid A)$= The probability of B given A

$P(A \cap B)$)= The probability of both A and B occurring

Applications of Bayes' theorem:

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC AUTONOMOUS)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD-501 401

ASSESSMENT OF PROGRAMME OUTCOMES & PROGRAMME SPECIFIC OUTCOMES

PROGRAMME **B.TECH (CSE)**

YEAR III SEM V Academic Year 2021-2022 ***BATCH*** ***2019-202***

Course Code A30530 Course Name ARTIFICAL INTELLIGENCE

ARTICULATION

S.No	COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1
1	**CO1**	3	1	2										1
2	**CO2**	2	3	2										1
3	**CO3**	3	2	1									1	1
4	**CO4**	3	2	2		2							2	2
5	**CO5**	3	1	3		2							1	
Average		3	1	2		2							2	1

FINAL ATTAINMENT (70% of External marks + 30% of Internal marks)				
Description	CO1	C02	C03	CO4
External Examinations Attainment	3.00	3.00	2.00	1.00
Internal Examinations Attainment	2.00	2.00	2.00	2.00
70% of External Examinations Attainment	2.10	2.10	1.40	0.70
30% of Internal Examinations	0.60	0.60	0.60	0.60
Final Attainment (70% of Ext + 30% of Int)	2.70	2.70	2.00	1.30

ATTAINMENT OF POs & PSOs THROUGH THE COURSE OUTCOMES

COs	Attainment	PO1	PO2	PO3	PO4	PO5	PO6	P07	PO8	PO9	PO10	PO11	PO12	PSO1
CO1	2.70	3	1	2										1
CO2	2.70	2	3	2										1
CO3	2.00	3	2	1									1	1
CO4	1.30	3	2	2		2							2	2
CO5	2.70	3	1	3		2							1	
Attainment		**2.25**	**2.23**	**2.35**	-	**2.00**	-	-	-	-	-	-	**1.83**	**2.00**

(Course Coordinator) (Programme Coordinator

BOOKLET NUMBER :	CSE-A College Stamp	R18

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

Kandlakoya, Medchal, Hyderabad - 501 401.

MID SEMESTER EXAMINATION ANSWER BOOK

Registered No. 2 1 H 5 1 A 0 5 0 5

FIRST / SECOND SEMESTER EXAMINATION B.Tech./M.Tech./MBA V Semester NOV 2023 (Month and year)

Subject : AI

Date : 3/11/2023

Signature of the Invigilator with date

INSTRUCTIONS TO THE CANDIDATES

1. This booklet contains 16 pages. Candidates must ensure it before writing and in case a defective answer book is issued it must be returned to the invigilator and a new and defect free booklet must be obtained.
2. Before the candidate begins to answer, registered number, particulars of year, semester, subject etc., are to be filled in. Failure to enter all or any of these particulars may disqualify the paper from valuation.
3. Candidate is prohibited from
   (a) Writing.
   ☞ anything addressing the examiner in any manner whatsoever, in their answer book.
   ☞ Objectionable/obscene language in the answer book.
   ☞ anything other than their Registered Number on the question paper.
   (b) either seeking or providing any assistance to the fellow candidates in the exam.
   (c) possessing a manuscript or a printed matter, in any form, in the examination hall.
   (d) bringing loose sheets or paper into the examination hall and detaching any paper from the answer book.
   (e) carrying Mobile Phone to Exam Hall.
   Violation of these instructions will be viewed as a case of malpractice, which is a punishable offence.
4. Before beginning to answer any question, candidates must write the correct question number, in the margin only and should not write anything else in the margin.
5. Answers must be written legibly on both sides of the paper. There shall be about 25 lines in each page. It is not necessary to begin each answer on a fresh page. Candidates should not use any other ink, except BLACK or BLUE ink.
6. Rough work, if any, must be separated, from the subject matter, by a line and noted as rough work.
7. The answer book, at the end of the examination, must be handed over to the Assistant Superintendent (Invigilator) by the candidate **This responsibility lies with the candidate only.**
8. Candidates should maintain absolute silence during the time of examination. Misbehavior, in any form, by the candidate, in the examination hall, will attract severe punishment.
9. Candidates are permitted to leave the examination hall only after the expiry of half of the allotted time and candidates will be permitted to carry the question paper only when they are leaving the exam hall in the last half-an-hour.
10. **No additional answer books will be supplied.**

To be filled in by the Examiner only

PART - A / PART - B

MARKS SLIP

PART-A	Q.No.	1	2	3	4	5	—	—	—	—	—	Part-A Total
	Marks	1	2	1	2	1						7
PART-B	Q.No.	6	7	8	—	—	—	—	—	—	—	Part-B Total
		A / B	A / B	A / B								
	Marks	5	4	5								14
Grand Total in Words : TWO ONE											GRAND TOTAL	21

Signature of the Scrutinizer with Date

P. Nags 7/11/23
Signature of the Examiner with Date

Part-B

6A:- A*

A* is a finding the solution form source to destination

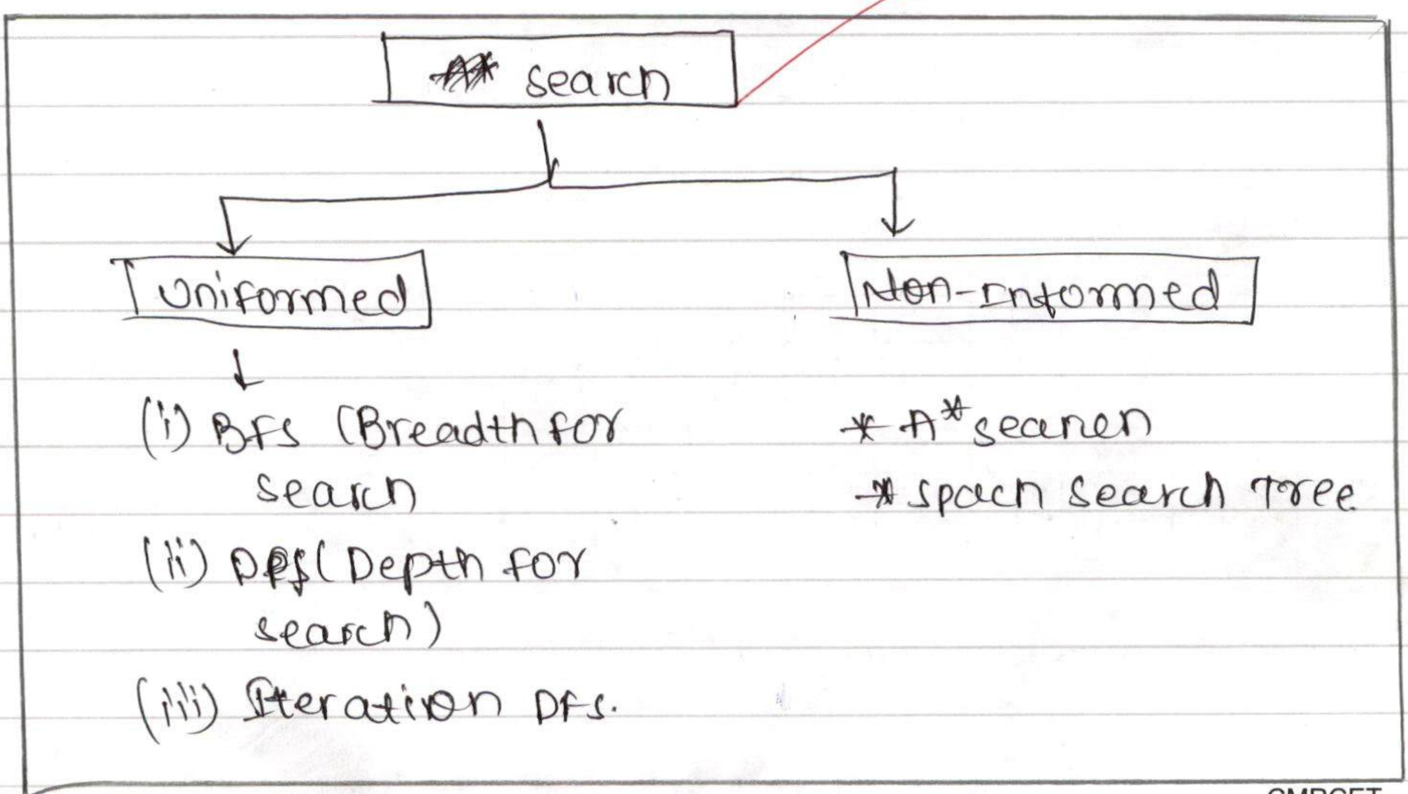
* It is a combination of BFS (Breadth for search) and DFS (Depth for search).

* The main is, It finds the shortest path.

* It finds the best solution for a node, which has smallest node, and continoues till reaches the final reach point.

* It finds the space search tree.

*



* It doesn't allow the backtracking.

* It is same of depends on BFS
* It gives results faster
*

$$f(n) = g(n) + h(n)$$

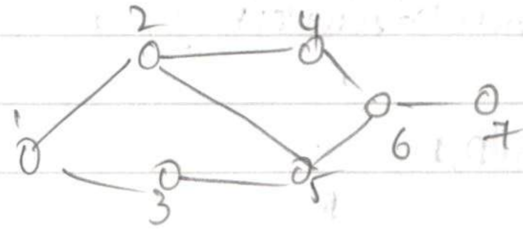
* It is same like hestric Algorithm

BFS (~~Breadth~~ first search)
* Start.
* step2: take the first node. ,check the node is adjacent values
* Step3: It moves to the next step.

Step4: again check the adjacent values, who is shorter value that node will continue the next.
* It performs on Queue first in first Out
* step5: After reach the destination there is no back tracking.

* Timecomplexity : O(n)
  Time takes = O(In)

Ex:



(i) | 1 | | | | |

(ii) take the I/P Q out.

| 0 | 2 | | | | 1 |

(iii) | | 2 | 4 | |

(iv) | | | 6 | |

(v) | | | 1 | 7 | ... |

(vi) | | | | | 1 |

It becomes untill Queue empty queue.

7A) wumpus world:

wumpus world is technique to the solve the the program or problem, to obtain Q find the best path

* It has Agent, Agent is anything the preciving the environment that acurators.

Stench.		Breeze	pit
wumpus evil	steanch gold	pit	Breeze
strench		Breeze	
	Breeze.	pit.	Breeze

Agent.

conditions :-

(i) In this wumpus world there so many in this

*) Agent
* Breeze
* Stench
* pit.
* wumpum

(ii) The pit has a adjacent ~~to~~, is there beside the Breeze.

(iii) Stench mean smell, the devil smell is adjacent to the wumpus.

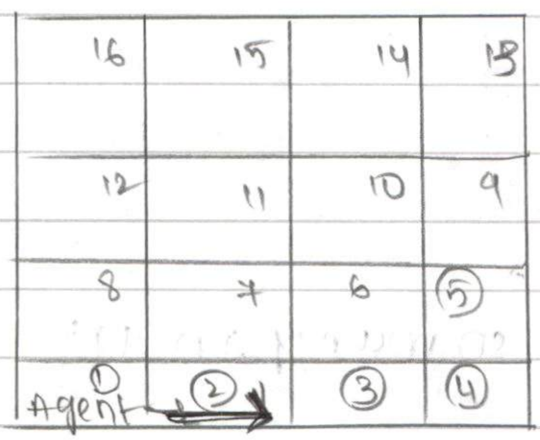
(iv) The Breeze smells the pit.

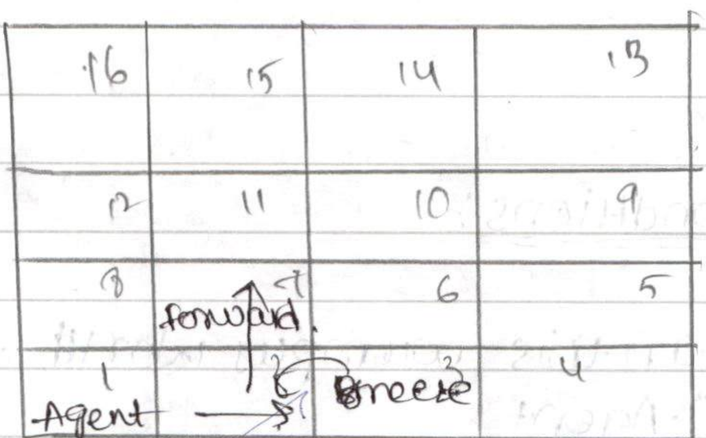
(i) In first we have an Agent, The Agent moves towards the forward direction.

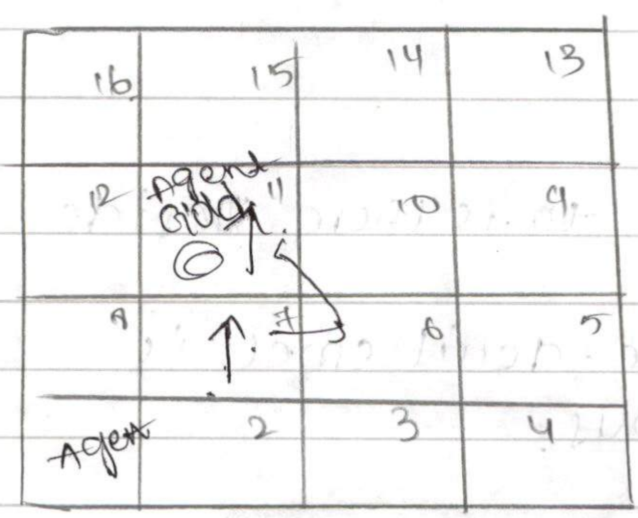
(ii) The Agent is identify the pit, pit there near the Breeze.
So, Agent doesn't move to the pit.

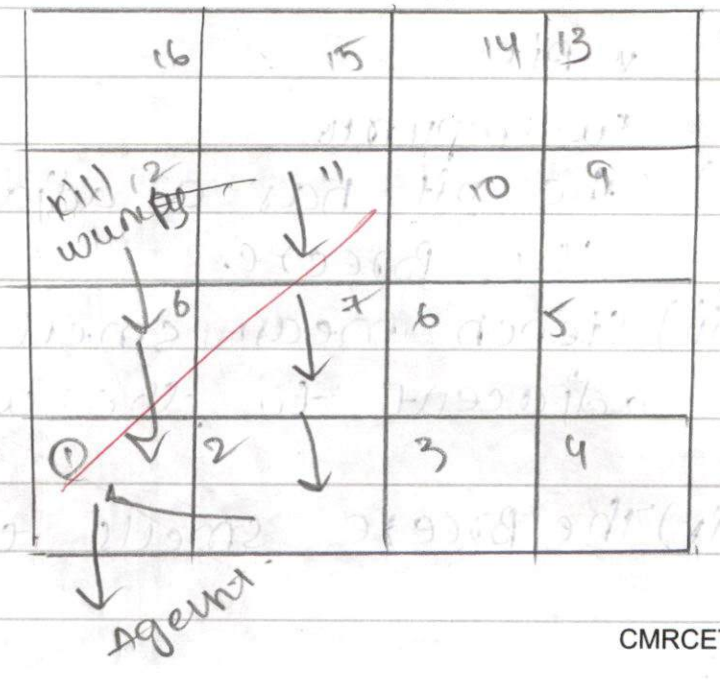
(iii) Agent moves the right right direction

(iv) It's also have the right - left direction in this wumpus world.









* Then Agent killed the wumpus the win the game

* He climbes the Gold again wampus had route the forward direction

* If the wumpus moves wrong direction, the wumpus will killed Agent.
* But we find a best path to kill wumpus.

* Then, the Breeze will Identify the Agent is near to the agent

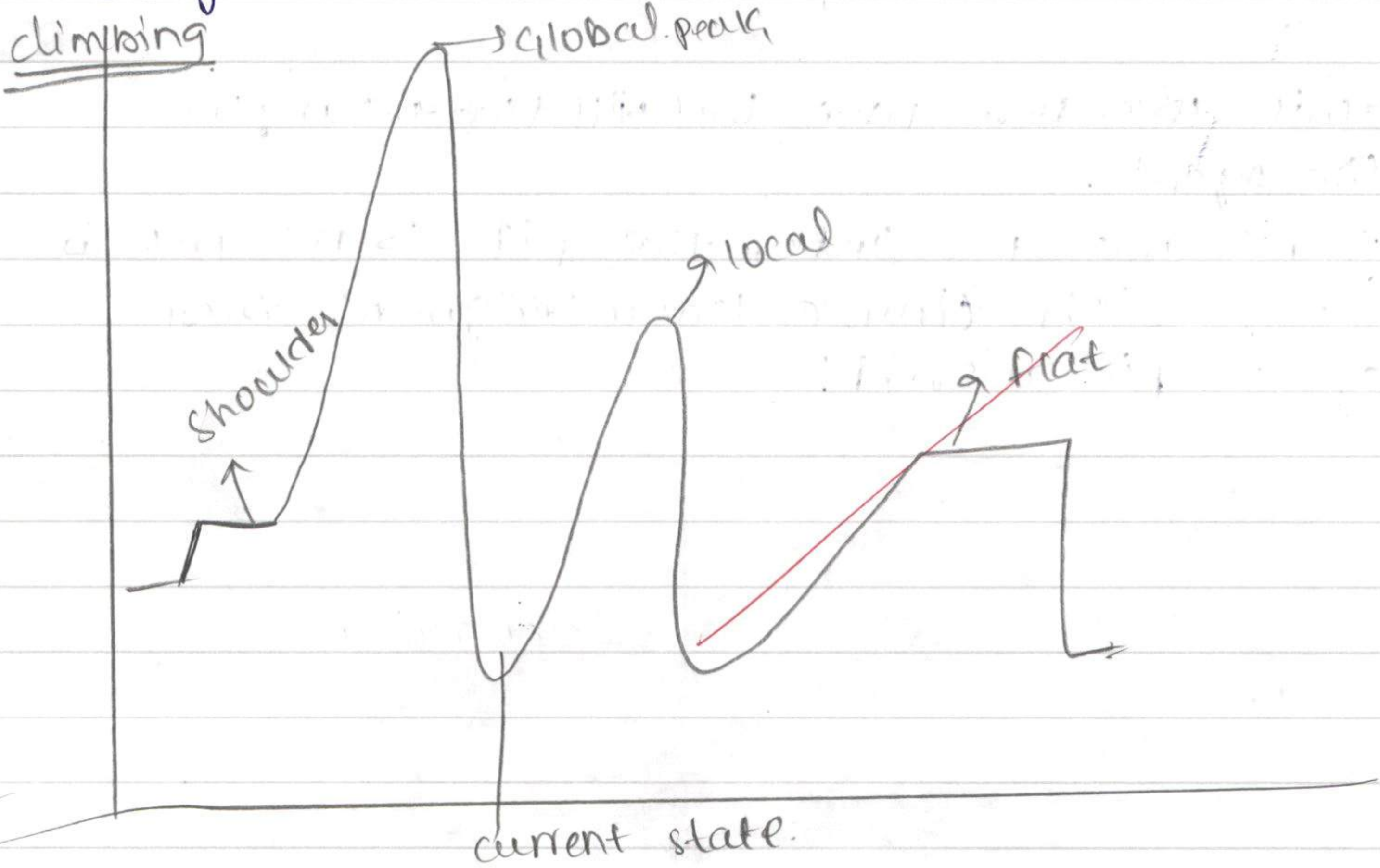
* This the best path to kill the wumpus the Agent.
* This the the infer the pit is no pit in [1,2] with simple knowledgment base of wumpus world:

(8B) Hill climbing Approach:

* Hill climbing Approach means, It's a high peak, It doesn't near the neighbour neart the peak path.
* It doesn't go the backtracking

* It doesn't allow the backtracking
  Ex: TSP (Travel sales person)

* Hill climbing is approach of easiet way but If doesn't back the same the way he came.



features
shoulder:

It is a starting of the hill climbing.

→ It will start the point from the first one.

Global climbing:

Global climbing is defined, It is highest peak of Hill climbing.

→ It doesn't have the current state & neighbour state.

→ It is very peak state

Local climbing:

Local is defined as a, It is between the ~~curren~~ peak state & flat state.

* flat:

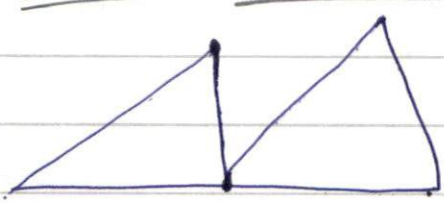
flat is ~~diff.~~ It is between the current state & neighbour state.

→ It doesn't want to go ~~It~~ is also knows same like TSP.

→ It has current states & neighbour state.

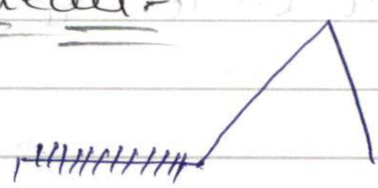
process:

(i) local maxima:



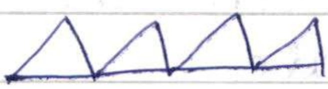
* It is highest peak of the any Hill

(ii) plateau:-



It is peak or any ground of the plateau

(iii) Ridges:



* continous like Hills.

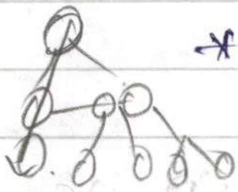
Types of Hill climbing:-

(i) simple Hill climbing
(ii) strench climbing
(iii) stocostic climbing.

(i) simple Hill climbing:
simple Hill climbing is nothing but it is b/w node to neighbour.

* It is simple way to find the solution.

* from one path to an next path of nodes



* It choose the easy way.

(ii) strench climbing:-
It choose the node, to find the shortest path from source to destination
→ It is choose one node to another node.

(iii) stocostic climbing:-
It choose the random nodes to find the path.
→ It is defined as stocostic climbing.

Part - A

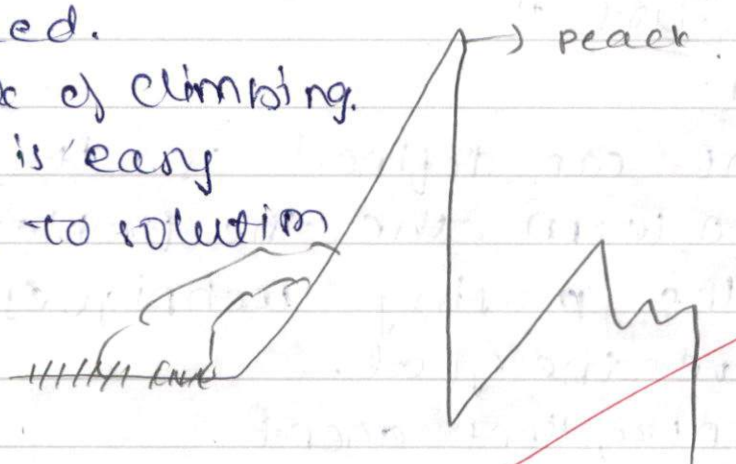
① Intelligent Agent:
Intellegent Agent can defined as, It
It is obtain to learn the the problem, and to solve the p using techniques find path, to acheive the goal.
* This is called Integellent agent.

2. predicate Logic:-
predicate Logic is also know as FOL (first order Logic.
* The relations between the object & qualities

first order Logic:
first order Logic is also known as Predicate Logic.
* It is powerfull language, the finds the the object, solutions for the objects & qualities

(5) simulated annealing search:
* simulated annealing search is a shorest path
* It gives results the faster.

* It finds the best solution speed of accerany of speed.
* peak of climbing.
* It is easy way to solution



3) CSP:-

constraint system problem is defined as the finds the solution.

* It has Domains.
  ↳ constraint
  ⊕

④ searching strategies:

* searching strategies comes under the BFS & DFS
* It finds the shortest path for a problem.
* The searching strategies are many uniformed & informed.

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

BOOKLET NUMBER :

College Stamp

R18

CMR GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

Kandlakoya, Medchal, Hyderabad - 501 401. CSE - A.

MID SEMESTER EXAMINATION ANSWER BOOK

Registered No. | 2 | 1 | H | 5 | 1 | A | 0 | 5 | 1 | 4 |

FIRST / SECOND SEMESTER EXAMINATION B.Tech./M.Tech./MBA II Semester Dec/2023 (Month and year)

Subject : AI

Date : 30/12/23.

Signature of the Invigilator with date 29/12/23

INSTRUCTIONS TO THE CANDIDATES

1. This booklet contains 16 pages. Candidates must ensure it before writing and in case a defective answer book is issued it must be returned to the invigilator and a new and defect free booklet must be obtained.
2. Before the candidate begins to answer, registered number, particulars of year, semester, subject etc., are to be filled in. Failure to enter all or any of these particulars may disqualify the paper from valuation.
3. Candidate is prohibited from
   (a) Writing.
   ☞ anything addressing the examiner in any manner whatsoever, in their answer book.
   ☞ Objectionable/obscene language in the answer book.
   ☞ anything other than their Registered Number on the question paper.
   (b) either seeking or providing any assistance to the fellow candidates in the exam.
   (c) possessing a manuscript or a printed matter, in any form, in the examination hall.
   (d) bringing loose sheets or paper into the examination hall and detaching any paper from the answer book.
   (e) carrying Mobile Phone to Exam Hall.
   Violation of these instructions will be viewed as a case of malpractice, which is a punishable offence.
4. Before beginning to answer any question, candidates must write the correct question number, in the margin only and should not write anything else in the margin.
5. Answers must be written legibly on both sides of the paper. There shall be about 25 lines in each page. It is not necessary to begin each answer on a fresh page. Candidates should not use any other ink, except BLACK or BLUE ink.
6. Rough work, if any, must be separated, from the subject matter, by a line and noted as rough work.
7. The answer book, at the end of the examination, must be handed over to the Assistant Superintendent (Invigilator) by the candidate **This responsibility lies with the candidate only.**
8. Candidates should maintain absolute silence during the time of examination. Misbehavior, in any form, by the candidate, in the examination hall, will attract severe punishment.
9. Candidates are permitted to leave the examination hall only after the expiry of half of the allotted time and candidates will be permitted to carry the question paper only when they are leaving the exam hall in the last half-an-hour.
10. **No additional answer books will be supplied.**

To be filled in by the Examiner only

PART - A / PART - B

MARKS SLIP

PART-A	Q.No.	1	2	3	4	5	—	—	—	—	—	**Part-A Total**
	Marks	1	0	1	1	1						04

PART-B	Q.No.	6		7		8		—	—	—	—	—	—	—	—	**Part-B Total**
		A	B	A	B	A	B									
	Marks		2		1		4									07

Grand Total in Words : One - One.	**GRAND TOTAL**	11

Signature of the Scrutinizer with Date

P. Nay 30/12/23.

Signature of the Examiner with Date

CMR GROUP OF INSTITUTIONS
EXPLORE TO INVENT

PART-A.

1. Ontological engineering involves certain explicit and formal Specification of the Concepts, entries, and relationship within a particular domain.

3: Contingent planning involves making flexible plans that can adapt to the environment.
ex:- Senraio: planning a route for a delevery truck in unpredectable weather condition.

4. Uncertainity refers to the lack of complete knowledge or predectiblity about a outcome of a certain event or Situation
Types:-
-) epistimic uncertainity.
-) Aleatory Uncertainity.

5.

Aspect	Supervise L.
-) Traning data	-) Labeled.
-) Objective.	-) predict Output
-) Example.	-) Image Classif
-) Feedback Mechanism	-) error Correct.

2. Classical planning. is a planing which is done certainly is known as classical planig.

0

Part - B

6B. Reasoning with default information
1) Default Assumption:- Default information consist of assumption that are considered true in the absence of evidence is known as Default of Assumption is available.
Ascepts of Default Assumptions:-
→ Default Assumption
→ Default rules.
→ Non-motion. objects. Logic.
~~Key~~ ~~Concept~~ ~~Of~~ reasoning
→ Relevance and Palusablity.
Example for reasoning with default with ~~on~~ information.
• Senario :- Default rule :- "Birds can only fly.

Situation :-

I. Instances 2 :-

- Observation :- "Robin can fly"
- explanation :- Default rule "Birds can fly" is applicable to the Birds robin cannot fly.

(8B) Bayes' Rule also known as Bayes therom. Or Bayes law is a fundamental principle in Probablity theory that discribe the knowledge of condition that might be. related to. event. based or prior knowledge condition, the formula for Bayes Rule is follows as:-

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

where, P(A/B) is the probablity of a event a given that event B is occured.
P(B/A) is a probablity of event B. given that event A is occured.
P(A) and P(B) are the probablity of event A and B.

Uses of Bayes Rule.
1. Medical. Diognastic:-
ex:- Suppose a medical test is conducted. to detect a dieasease and the test has known as false positive rate and false negative. rate. Bayes's therom can be used to update the probablity of. having the dease based on the result

2. Spam filtering:-
example:- In email Spam filtering the Bay's Rule is used to calculate the probablity that an email is spam given certain words or features. improve the accurcy of spam detection by updateing.

3. Weather Prediction:-
example:- Baye's rule applied in weather prediction to update the probablity of rain given to certain meterlogical conditions. it helps metrological makes. mor accurate prediction.

4. Mechain Learning:-
Example:- The Mechain Learning Bays law is used it classify the probablity of Mechain learning is the main use in Mechain Learning.

(7B.) planning with State Space Search involves exploring a State space to find a sequence or action that trans-form an initial state to a goal State. various algorithms are employed for the purpose Here a explanation of differnt algorithm used for State-space

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

1. Depth - Frist Search (DFS) :-

- principle :- explorer as far as possible along a branch before back tracking.
- Implementation :- utilizes a stack to keep on the track of the exploration path.
- pro's and Cons :- Pros :-
  → Memory efficent, especially in senaraio's with deep branching.
  - Cons :- May not find the optimal solution and it maight get stuck in deep path.

2. Bredth - Frist Search (BFS) :-

- principle :- explorer all neighbour nodes at the current depth before moving on the nodes at the next depth level.
- Implementation :- utilizes a queue to explore nodes in a bredth - frist element.
- Pros and Cons :- pros :-
  → ~~duration~~ guarantee finding the optimal. solution with the forecast action
  - Cons :- Memory intensive, particularly in scenario with band branching.

Nays

UNIT - I

Introduction to A.I

Artificial Intelligence is composed of two words Artificial and Intelligence, where Artificial defines "man made," and intelligence defines "thinking power", hence AI means "a man-made thinking power."

So, we can define AI as:

"It is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions."

Artificial Intelligence exists when a machine can have human-based skills such as learning, reasoning, and solving problems. With Artificial Intelligence you do not need to pre-program a machine to do some work, despite that you can create a machine with programmed algorithms which can work with own intelligence, and that is the awesomeness of AI.

Definitions of A.I

Thinking Humanly "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)	**Thinking Rationally** "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)
Acting Humanly "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)	**Acting Rationally** "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)

Figure 1.1 Some definitions of artificial intelligence, organized into four categories.

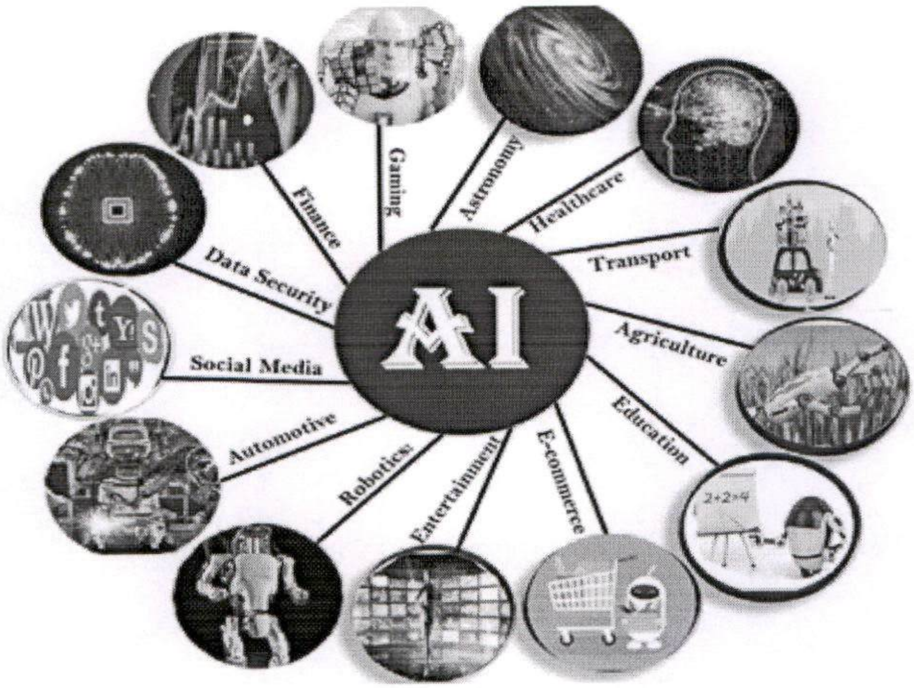
	Human- Like	Rationally
Think:	Cognitive Science Approach *"Machines that think like humans"*	Laws of thought Approach *" Machines that think Rationally"*
Act:	Turing Test Approach *"Machines that behave like humans"*	Rational Agent Approach *"Machines that behave Rationally"*

Goals of Artificial Intelligence

Following are the main goals of Artificial Intelligence:

- Replicate human intelligence
- Solve Knowledge-intensive tasks
- An intelligent connection of perception and action
- Building a machine which can perform tasks that requires human intelligence such as:
    - Proving a theorem
    - Playing chess
    - Plan some surgical operation
    - Driving a car in traffic
- Creating some system which can exhibit intelligent behaviour, learn new things by itself, demonstrate, explain, and can advise to its user.

Application of AI



1) AI in Astronomy

- Artificial Intelligence can be very useful to solve complex universe problems. AI technology can be helpful for understanding the universe such as how it works, origin, etc. 2. AI in Healthcare
- In the last, five to ten years, AI becoming more advantageous for the healthcare industry and going to have a significant impact on this industry.
- Healthcare Industries are applying AI to make a better and faster diagnosis than humans. AI can help doctors with diagnoses and can inform when patients are worsening so that medical help can reach to the patient before hospitalization.

3. AI in Gaming

- AI can be used for gaming purpose. The AI machines can play strategic games like chess, where the machine needs to think of a large number of possible places.

4. AI in Finance

- AI and finance industries are the best matches for each other. The finance industry is implementing automation, chatbot, adaptive intelligence, algorithm trading, and machine learning into financial processes.

5. AI in Data Security
   - The security of data is crucial for every company and cyber-attacks are growing very rapidly in the digital world. AI can be used to make your data more safe and secure. Some examples such as AEG bot, AI2 Platform, are used to determine software bug and cyber-attacks in a better way.

6. AI in social media
   - Social Media sites such as Facebook, Twitter, and Snapchat contain billions of user profiles, which need to be stored and managed in a very efficient way. AI can organize and manage massive amounts of data. AI can analyse lots of data to identify the latest trends, hashtag, and requirement of different users.
7. AI in Travel & Transport
   - AI is becoming highly demanding for travel industries. AI is capable of doing various travel related works such as from making travel arrangement to suggesting the hotels, flights, and best routes to the customers. Travel industries are using AI powered chatbots which can make human-like interaction with customers for better and fast response.

8. AI in Automotive Industry
   - Some Automotive industries are using AI to provide virtual assistant to their user for better performance. Such as Tesla has introduced Tesla Bot, an intelligent virtual assistant.

   - Various Industries are currently working for developing self-driven cars which can make your journey more safe and secure.

9. AI in Robotics:
   - Artificial Intelligence has a remarkable role in Robotics. Usually, general robots are programmed such that they

can perform some repetitive task, but with the help of AI, we can create intelligent robots which can perform tasks with their own experiences without pre-programmed.

- Humanoid Robots are best examples for AI in robotics, recently the intelligent Humanoid robot named as Erica and Sophia has been developed which can talk and behave like humans.

10. AI in Entertainment

- We are currently using some AI based applications in our daily life with some entertainment services such as Netflix or Amazon. With the help of ML/AI algorithms, these services show the recommendations for programs or shows.

11. AI in Agriculture

Agriculture is an area which requires various resources, labour, money, and time for best result. Now a day's agriculture is becoming digital, and AI is emerging in this field. Agriculture is applying AI as agriculture robotics, solid and crop monitoring, predictive analysis. AI in agriculture can be very helpful for farmers.

12. AI in E-commerce

- AI is providing a competitive edge to the e-commerce industry, and it is becoming more demanding in the e-commerce business. AI is helping shoppers to discover associated products with recommended size, color, or even brand.

13. AI in education:

- AI can automate grading so that the tutor can have more time to teach. AI chatbot can communicate with students as a teaching assistant.

- AI in the future can be work as a personal virtual tutor for students, which will be accessible easily at any time and any place.

History of Artificial Intelligence

- 1946: ENIAC was first electronic general purpose computers were developed.
- 1950 : Alan Turing asks the question **"CAN MACHINES THINK ?"**
- **1956 :** Dartmouth meeting: "Artificial Intelligence" adopted
- **Founders of A.I**

  McCarthy

  Marvin Minsky

  Alan Newell

  Herbert Simon

- 1950**:** Turing Test for Machine Intelligence
- 1956 : AI born at Dartmouth Collage workshop
- 1964 : Eliza- Chatbot Psychotherapist
- 1966 : First general purpose mobile robot named Shakey was developed.
- 1974 -1980 : Winter (Failure of A.I)

  - Failure of machine translation

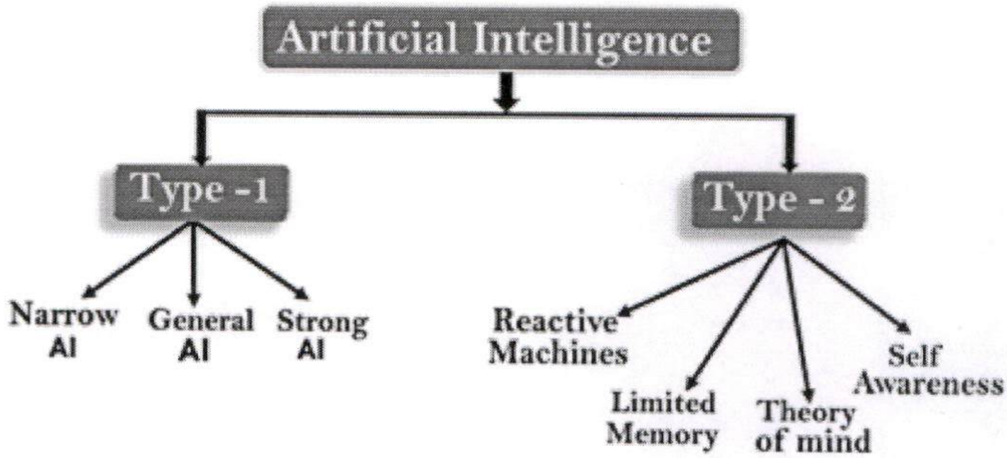
  - There were negative results in Neural nets.

  - Poor speech understanding

- Mid 90's: Cortes and Vapnik published paper on Support Vector Machines

Types of Artificial Intelligence:

Artificial Intelligence can be divided in various types, there are mainly two types of main categorization which are based on capabilities and based on functionally of AI. Following is flow diagram which explains the types of AI.



AI type-1: Based on Capabilities

1. Weak AI or Narrow AI:

- Narrow AI is a type of AI which is able to perform a dedicated task with intelligence. The most common and currently available AI is Narrow AI in the world of Artificial Intelligence.
- Narrow AI cannot perform beyond its field or limitations, as it is only trained for one specific task. Hence it is also termed as weak AI. Narrow AI can fail in unpredictable ways if it goes beyond its limits.
- Apple Siri is a good example of Narrow AI, but it operates with a limited pre-defined range of functions.
- IBM's Watson supercomputer also comes under Narrow AI, as it uses an Expert system approach combined with Machine learning and natural language processing.
- Some Examples of Narrow AI are playing chess, purchasing suggestions on e-commerce site, self-driving cars, speech recognition, and image recognition.

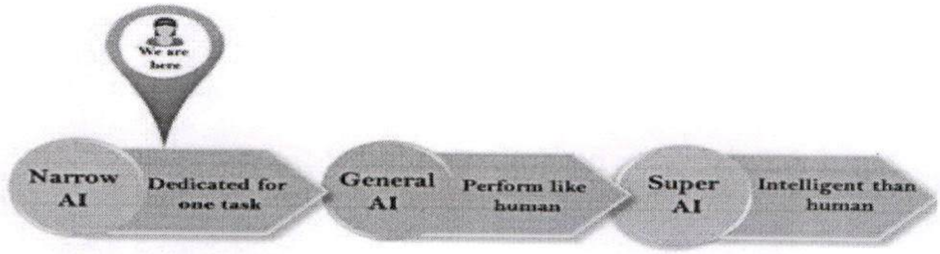
2. General AI:

- General AI is a type of intelligence which could perform any intellectual task with efficiency like a human.
- The idea behind the general AI to make such a system which could be smarter and think like a human by its own.
- Currently, there is no such system exist which could come under general AI and can perform any task as perfect as a human.

- The worldwide researchers are now focused on developing machines with General AI.
- As systems with general AI are still under research, and it will take lots of efforts and time to develop such systems.

3. **Super AI:**

- Super AI is a level of Intelligence of Systems at which machines could surpass human intelligence, and can perform any task better than human with cognitive properties. It is an outcome of general AI.
- Some key characteristics of strong AI include capability include the ability to think, to reason, solve the puzzle, make judgments, plan, learn, and communicate by its own.
- Super AI is still a hypothetical concept of Artificial Intelligence. Development of such systems in real is still world changing task.



Artificial Intelligence type-2:

Based on functionality

1. Reactive Machines
   - Purely reactive machines are the most basic types of Artificial Intelligence.
   - Such AI systems do not store memories or past experiences for future actions.
   - These machines only focus on current scenarios and react on it as per possible best action.
   - IBM's Deep Blue system is an example of reactive machines. o Google's AlphaGo is also an example of reactive machines.
2. Limited Memory

- Limited memory machines can store past experiences or some data for a short period of time.
- These machines can use stored data for a limited time period only.
- Self-driving cars are one of the best examples of Limited Memory systems. These cars can store recent speed of nearby cars, the distance of other cars, speed limit, and other information to navigate the road.

3. Theory of Mind

    - Theory of Mind AI should understand the human emotions, people, beliefs, and be able to interact socially like humans.
    - This type of AI machines is still not developed, but researchers are making lots of efforts and improvement for developing such AI machines.

4. Self-Awareness

    - Self-awareness AI is the future of Artificial Intelligence. These machines will be super intelligent, and will have their own consciousness, sentiments, and self-awareness.
    - These machines will be smarter than human mind.
    - Self-Awareness AI does not exist in reality still and it is a hypothetical concept.

Agents: An agent is anything that can be viewed as **perceiving its environment through sensors** and acting upon that environment through **actuators**.

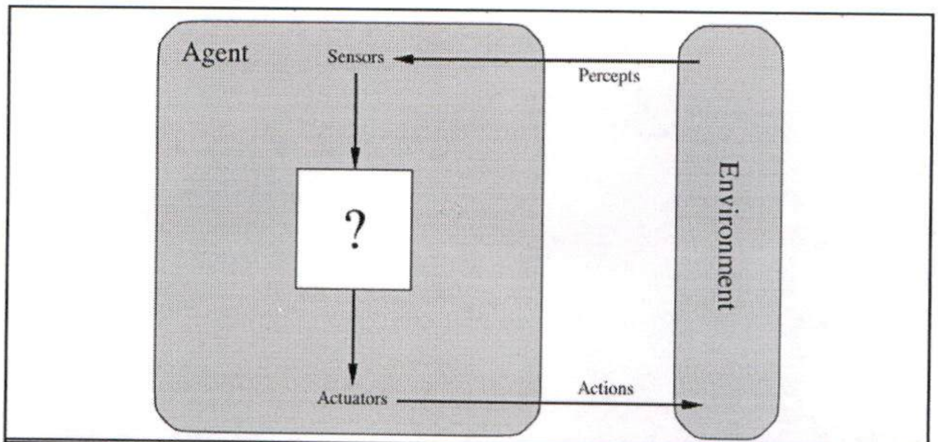


Figure 2.1 Agents interact with environments through sensors and actuators.

- Human Agents: Eyes, Nose, Mouth and other organs for sensors and hands, legs, vocal tract, and so on for actuators.
- Robotic Agent: Cameras and infrared range finders for sensors various motors for actuators.

- Mathematically speaking, we say that an agent's behavior is described by the agent function that maps any given percept sequence to an action.
- The agent function maps from percept histories to actions:

[*f*: *P** à *A*]

- The agent program runs on the physical architecture to produce *f*

Agent = Architecture + program

<u>Example—the vacuum-cleaner world</u>

- **This particular world has just two locations: squares A and B. The vacuum agent perceives which square it is in and whether there is dirt in the square. It can choose to move left, move right, suck up the dirt, or do nothing. One very simple agent function is the following: if the current square is dirty, then suck; otherwise, move to the other square.**

Unit 2

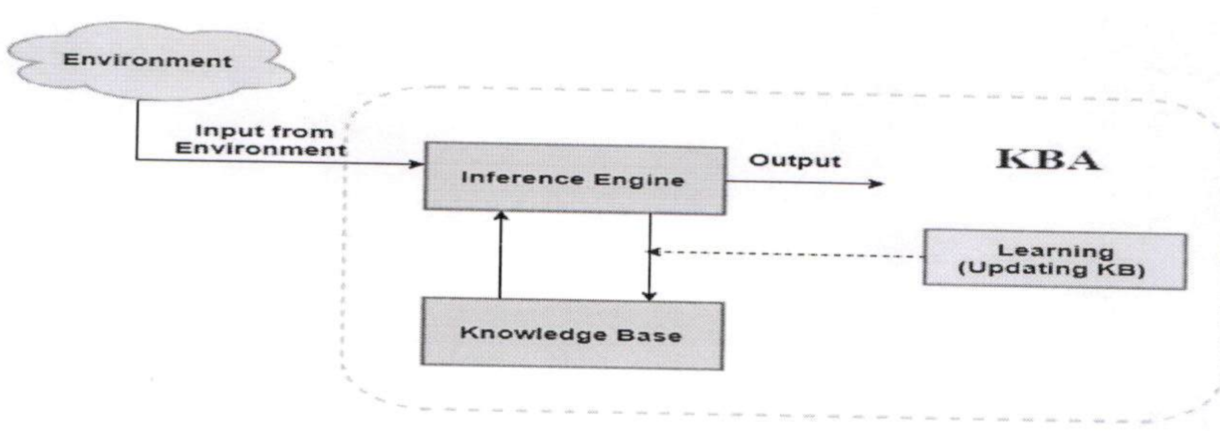
Knowledge-Based Agent in Artificial intelligence

- An intelligent agent needs **knowledge** about the real world for taking decisions and **reasoning** to act efficiently.
- Knowledge-based agents are those agents who have the capability of **maintaining an internal state of knowledge, reason over that knowledge, update their knowledge after observations and take actions. These agents can represent the world with some formal representation and act intelligently**.
- Knowledge-based agents are composed of two main parts:
    - **Knowledge-base and**
    - **Inference system.**

A knowledge-based agent must able to do the following:

- An agent should be able to represent states, actions, etc.
- An agent Should be able to incorporate new percepts
- An agent can update the internal representation of the world
- An agent can deduce the internal representation of the world
- An agent can deduce appropriate actions.

The architecture of knowledge-based agent:



The above diagram is representing a generalized architecture for a knowledge-based agent. The knowledge-based agent (KBA) takes input from the environment by perceiving the environment. The input is taken by the inference engine of the agent and which also communicate with KB to decide as per the knowledge store in KB. The learning element of KBA regularly updates the KB by learning new knowledge.

Knowledge base: Knowledge-base is a central component of a knowledge-based agent, it is also known as KB. It is a collection of sentences (here 'sentence' is a technical term and it is not identical to sentence in English). These sentences are expressed in a language which is called a knowledge representation language. The Knowledge-base of KBA stores fact about the world.

Why use a knowledge base?

Knowledge-base is required for updating knowledge for an agent to learn with experiences and take action as per the knowledge.

Inference system

Inference means deriving new sentences from old. Inference system allows us to add a new sentence to the knowledge base. A sentence is a proposition about the world. Inference system applies logical rules to the KB to deduce new information.

Inference system generates new facts so that an agent can update the KB. An inference system works mainly in two rules which are given as:

- **Forward chaining**
- **Backward chaining**

Operations Performed by KBA

Following are three operations which are performed by KBA in order to show the intelligent behavior:

1. **TELL:** This operation tells the knowledge base what it perceives from the environment.
2. **ASK:** This operation asks the knowledge base what action it should perform.
3. **Perform:** It performs the selected action.

A generic knowledge-based agent:

Following is the structure outline of a generic knowledge-based agents program:

```
function KB-AGENT(percept):
persistent: KB, a knowledge base
        t, a counter, initially 0, indicating time
TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
Action = ASK(KB, MAKE-ACTION-QUERY(t))
TELL(KB, MAKE-ACTION-SENTENCE(action, t))
t = t + 1
return action
```

The knowledge-based agent takes percept as input and returns an action as output. The agent maintains the knowledge base, KB, and it initially has some background knowledge of the real world. It also has a counter to indicate the time for the whole process, and this counter is initialized with zero.

Each time when the function is called, it performs its three operations:

- Firstly it TELLs the KB what it perceives.
- Secondly, it asks KB what action it should take
- Third agent program TELLS the KB that which action was chosen.

The MAKE-PERCEPT-SENTENCE generates a sentence as setting that the agent perceived the given percept at the given time.

The MAKE-ACTION-QUERY generates a sentence to ask which action should be done at the current time.

MAKE-ACTION-SENTENCE generates a sentence which asserts that the chosen action was executed.

Various levels of knowledge-based agent:

A knowledge-based agent can be viewed at different levels which are given below:

1. Knowledge level

Knowledge level is the first level of knowledge-based agent, and in this level, we need to specify what the agent knows, and what the agent goals are. With these specifications, we can fix its behavior. For example, suppose an automated taxi agent needs to go from a station A to station B, and he knows the way from A to B, so this comes at the knowledge level.

2. Logical level:

At this level, we understand that how the knowledge representation of knowledge is stored. At this level, sentences are encoded into different logics. At the logical level, an encoding of knowledge into logical sentences occurs. At the logical level we can expect to the automated taxi agent to reach to the destination B.

3. Implementation level:

This is the physical representation of logic and knowledge. At the implementation level agent perform actions as per logical and knowledge level. At this level, an automated taxi agent actually implements his knowledge and logic so that he can reach to the destination.

Approaches to designing a knowledge-based agent:

There are mainly two approaches to build a knowledge-based agent:

1. **1. Declarative approach:** We can create a knowledge-based agent by initializing with an empty knowledge base and telling the agent all the sentences with which we want to start with. This approach is called Declarative approach.
2. **2. Procedural approach:** In the procedural approach, we directly encode desired behavior as a program code. Which means we just need to write a program that already encodes the desired behavior or agent.

However, in the real world, a successful agent can be built by combining both declarative and procedural approaches, and declarative knowledge can often be compiled into more efficient procedural code.

What is knowledge representation?

Humans are best at understanding, reasoning, and interpreting knowledge. Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world. **But how machines do all these things comes under knowledge representation and reasoning**. Hence we can describe Knowledge representation as following:

- Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behavior of agents.
- It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language.
- It is also a way which describes how we can represent knowledge in artificial intelligence. Knowledge representation is not just storing data into some database, but it also enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human.

What to Represent:

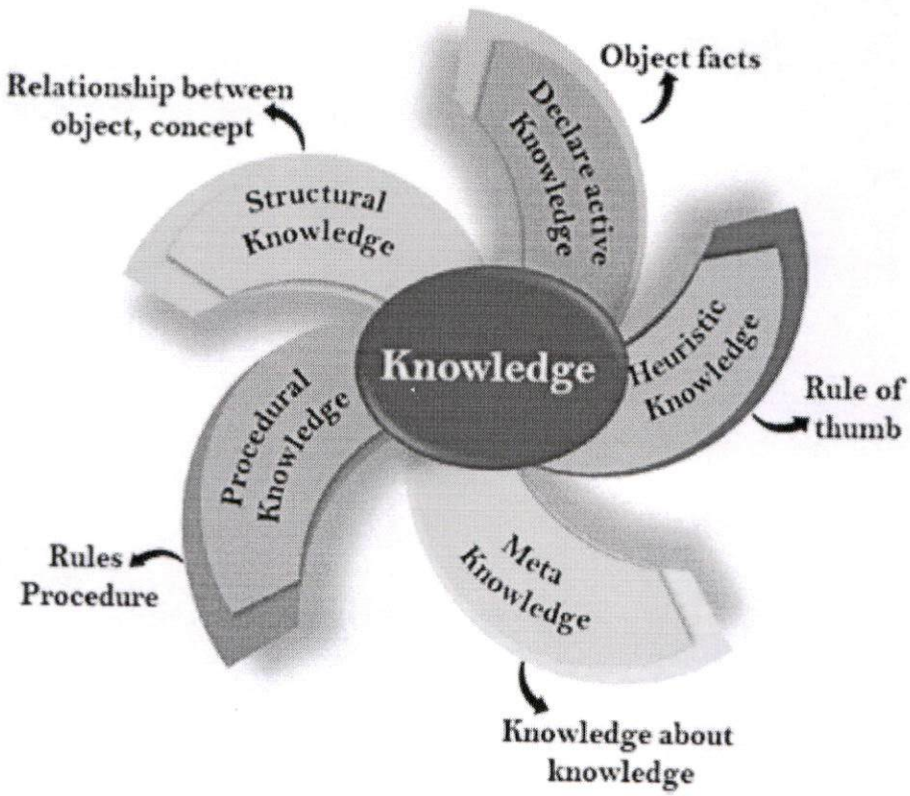
Following are the kind of knowledge which needs to be represented in AI systems:

- **Object:** All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments.
- **Events:** Events are the actions which occur in our world.
- **Performance:** It describe behavior which involves knowledge about how to do things.
- **Meta-knowledge:** It is knowledge about what we know.
- **Facts:** Facts are the truths about the real world and what we represent.
- **Knowledge-Base:** The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).

Knowledge: Knowledge is awareness or familiarity gained by experiences of facts, data, and situations. Following are the types of knowledge in artificial intelligence:

Types of knowledge

Following are the various types of knowledge:



1. Declarative Knowledge:

- Declarative knowledge is to know about something.
- It includes concepts, facts, and objects.
- It is also called descriptive knowledge and expressed in declarativesentences.
- It is simpler than procedural language.

2. Procedural Knowledge

- It is also known as imperative knowledge.

- Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.
- It can be directly applied to any task.
- It includes rules, strategies, procedures, agendas, etc.
- Procedural knowledge depends on the task on which it can be applied.

3. Meta-knowledge:

- Knowledge about the other types of knowledge is called Meta-knowledge.

4. Heuristic knowledge:

- Heuristic knowledge is representing knowledge of some experts in a filed or subject.
- Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.

5. Structural knowledge:

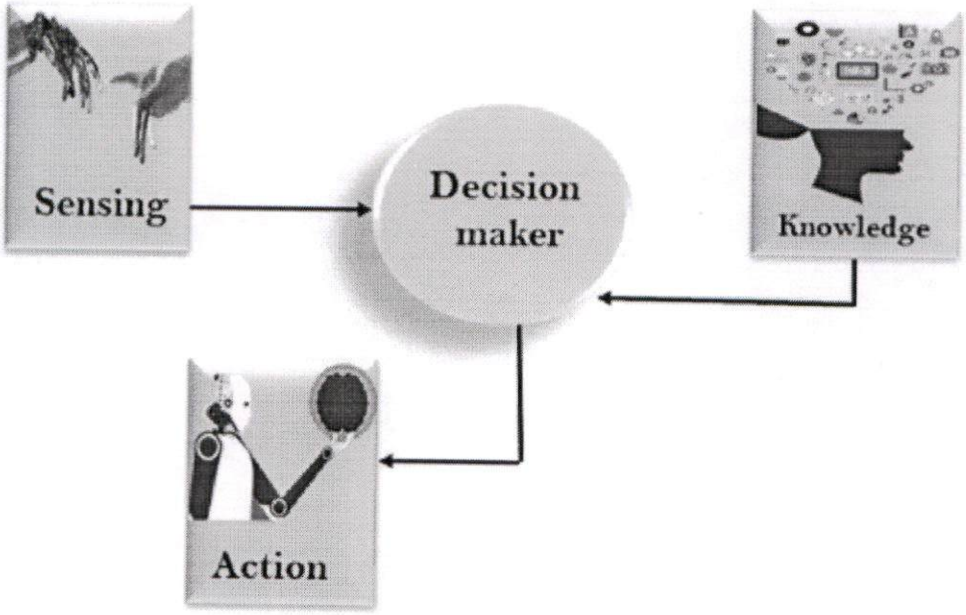
- Structural knowledge is basic knowledge to problem-solving.
- It describes relationships between various concepts such as kind of, part of, and grouping of something.
- It describes the relationship that exists between concepts or objects.

The relation between knowledge and intelligence:

Knowledge of real-worlds plays a vital role in intelligence and same for creating artificial intelligence. Knowledge plays an important role in demonstrating intelligent behavior in AI agents. An agent is only able to accurately act on some input when he has some knowledge or experience about that input.

Let's suppose if you met some person who is speaking in a language which you don't know, then how you will able to act on that. The same thing applies to the intelligent behavior of the agents.

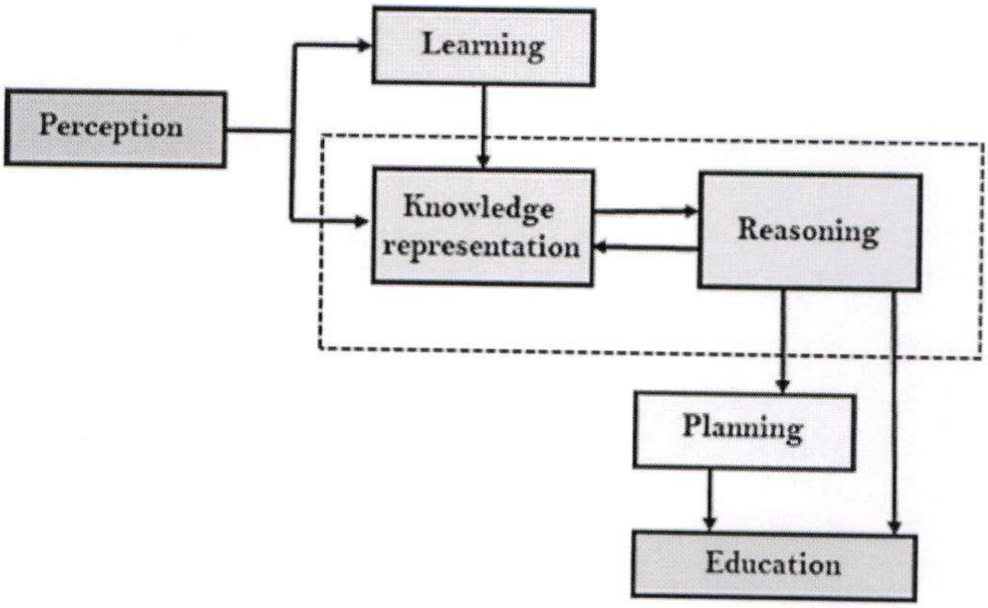
As we can see in below diagram, there is one decision maker which act by sensing the environment and using knowledge. But if the knowledge part will not present then, it cannot display intelligent behavior.



AI knowledge cycle:

An Artificial intelligence system has the following components for displaying intelligent behavior:

- Perception
- Learning
- Knowledge Representation and Reasoning
- Planning
- Execution



The above diagram is showing how an AI system can interact with the real world and what components help it to show intelligence. AI system has Perception component by which it retrieves information from its environment. It can be visual, audio or another form of sensory input. The learning component is responsible for learning from data captured by Perception comportment. In the complete cycle, the main components are knowledge representation and Reasoning. These two components are involved in showing the intelligence in machine-like humans. These two components are independent with each other but also coupled together. The planning and execution depend on analysis of Knowledge representation and reasoning.

Approaches to knowledge representation:

There are mainly four approaches to knowledge representation, which are givenbelow:

1. Simple relational knowledge:

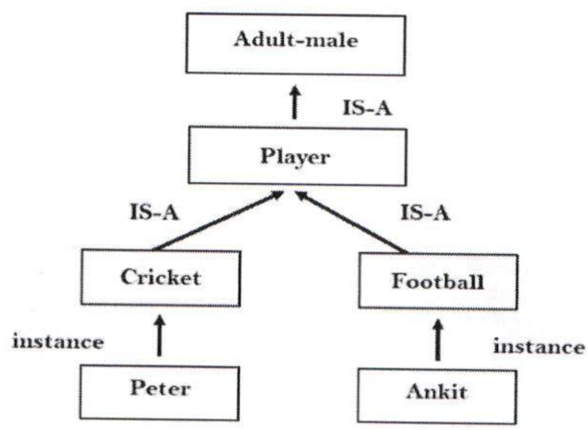
- It is the simplest way of storing facts which uses the relational method, and each fact about a set of the object is set out systematically in columns.
- This approach of knowledge representation is famous in database systems where the relationship between different entities is represented.
- This approach has little opportunity for inference.

Example: The following is the simple relational knowledge representation.

Player	Weight	Age
Player1	65	23
Player2	58	18
Player3	75	24

2. Inheritable knowledge:

- In the inheritable knowledge approach, all data must be stored into a hierarchy of classes.
- All classes should be arranged in a generalized form or a hierarchal manner.
- In this approach, we apply inheritance property.
- Elements inherit values from other members of a class.
- This approach contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation.
- Every individual frame can represent the collection of attributes and its value.
- In this approach, objects and values are represented in Boxed nodes.
- We use Arrows which point from objects to their values.
- **Example:**



Unit 3

Probabilistic reasoning in Artificial intelligence

Uncertainty:

Till now, we have learned knowledge representation using first-order logic and propositional logic with certainty, which means we were sure about the predicates. With this knowledge representation, we might write A→B, which means if A is true then B is true, but consider a situation where we are not sure about whether A is true or not then we cannot express this statement, this situation is called uncertainty.

So to represent uncertain knowledge, where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning.

Causes of uncertainty:

Following are some leading causes of uncertainty to occur in the real world.

1. Information occurred from unreliable sources.
2. Experimental Errors
3. Equipment fault
4. Temperature variation
5. Climate change.

Probabilistic reasoning:

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.

We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of someone's laziness and ignorance.

In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today," "behavior of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

Need of probabilistic reasoning in AI:

- When there are unpredictable outcomes.
- When specifications or possibilities of predicates becomes too large to handle.
- When an unknown error occurs during an experiment.

In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

- **Bayes' rule**
- **Bayesian Statistics**

Note: We will learn the above two rules in later chapters.

As probabilistic reasoning uses probability and related terms, so before understanding probabilistic reasoning, let's understand some common terms:

Probability: Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

1. $0 \leq P(A) \leq 1$, where P(A) is the probability of an event A.

1. $P(A) = 0$, indicates total uncertainty in an event A.

1. P(A) =1, indicates total certainty in an event A.

We can find the probability of an uncertain event by using the below formula.

$$\textbf{Probability of occurrence} = \frac{\textbf{Number of desired outcomes}}{\textbf{Total number of outcomes}}$$

- P(¬A) = probability of a not happening event.
- P(¬A) + P(A) = 1.

Event: Each possible outcome of a variable is called an event.

Sample space: The collection of all possible events is called sample space.

Random variables: Random variables are used to represent the events and objects in the real world.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior Probability: The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

Conditional probability:

Conditional probability is a probability of occurring an event when another event has already happened.

Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

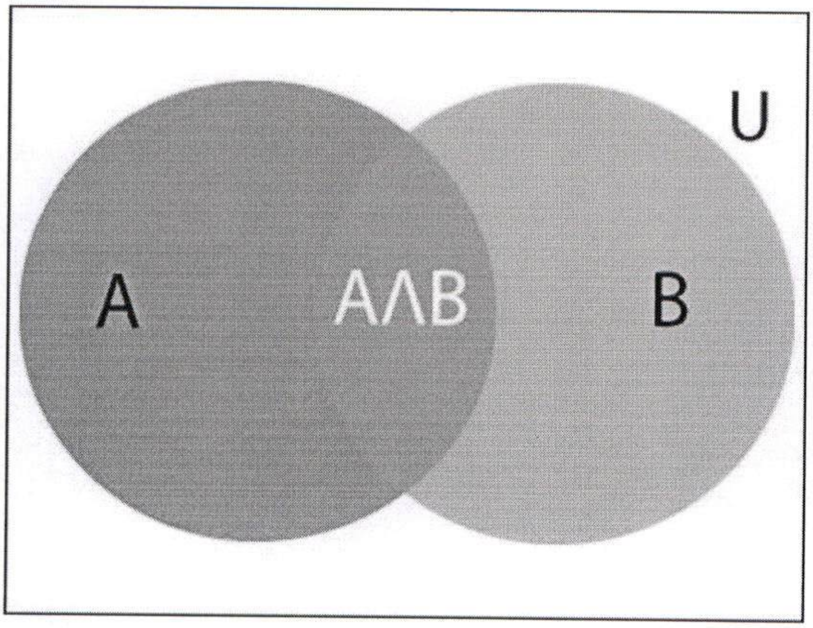
Where P($A \wedge B$)= Joint probability of a and B

P(B)= Marginal probability of B.

If the probability of A is given and we need to find the probability of B, then it will be given as:

$$P(B|A) = \frac{P(A \wedge B)}{P(A)}$$

It can be explained by using the below Venn diagram, where B is occurred event, so sample space will be reduced to set B, and now we can only calculate event A when event B is already occurred by dividing the probability of **P(A∧B) by P(B)**.



Example:

In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like English also like mathematics?

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{0.4}{0.7} = 57\%$$

Hence, 57% are the students who like English also like Mathematics.

Bayes' theorem in Artificial intelligence

Bayes' theorem:

Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.

Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.

It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).

Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Example: If cancer corresponds to one's age then by using Bayes' theorem, we can determine the probability of cancer more accurately with the help of age.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:

As from product rule we can write:

1. $P(A \wedge B) = P(A|B) P(B)$ or

Similarly, the probability of event B with known event A:

1. $P(A \wedge B) = P(B|A) P(A)$

Equating right hand side of both the equations, we will get:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad \text{....(a)}$$

The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

It shows the simple relationship between joint and conditional probabilities. Here,

P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

P(B|A) is called the likelihood, in which we consider that hypothesis is true, then we calculate the probability of evidence.

P(A) is called the **prior probability**, probability of hypothesis before considering the evidence

P(B) is called **marginal probability**, pure probability of an evidence.

In the equation (a), in general, we can write P (B) = P(A)*P(B|Ai), hence the Bayes' rule can be written as:

$$P(A_i|B) = \frac{P(A_i)*P(B|A_i)}{\sum_{i=1}^{k} P(A_i)*P(B|A_i)}$$

Where $A_1, A_2, A_3, \ldots\ldots, A_n$ is a set of mutually exclusive and exhaustive events.

Applying Bayes' rule:

Bayes' rule allows us to compute the single term P(B|A) in terms of P(A|B), P(*B*), and P(A). This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

$$\mathbf{P(cause|effect)} = \frac{P(effect|cause)\ P(cause)}{P(effect)}$$

Example-1:

Question: what is the probability that a patient has diseases meningitis with a stiff neck?

Given Data:

A doctor is aware that disease meningitis causes a patient to have a stiff neck, and it occurs 80% of the time. He is also aware of some more facts, which are given as follows:

- The Known probability that a patient has meningitis disease is 1/30,000.
- The Known probability that a patient has a stiff neck is 2%.

Let a be the proposition that patient has stiff neck and b be the proposition that patient has meningitis. , so we can calculate the following as:

P(a|b) = 0.8

P(b) = 1/30000

P(a)= .02

$$\mathbf{P(b|a)} = \frac{P(a|b)P(b)}{P(a)} = \frac{0.8*(\frac{1}{30000})}{0.02} = 0.001333333.$$

Hence, we can assume that 1 patient out of 750 patients has meningitis disease with a stiff neck.

Example-2:

Question: From a standard deck of playing cards, a single card is drawn. The probability that the card is king is 4/52, then calculate posterior probability P(King|Face), which means the drawn face card is a king card.

Solution:

$$\mathbf{P(king|face)} = \frac{P(Face|king)*P(King)}{P(Face)} \quad ....(i)$$

P(king): probability that the card is King= 4/52= 1/13

P(face): probability that a card is a face card= 3/13

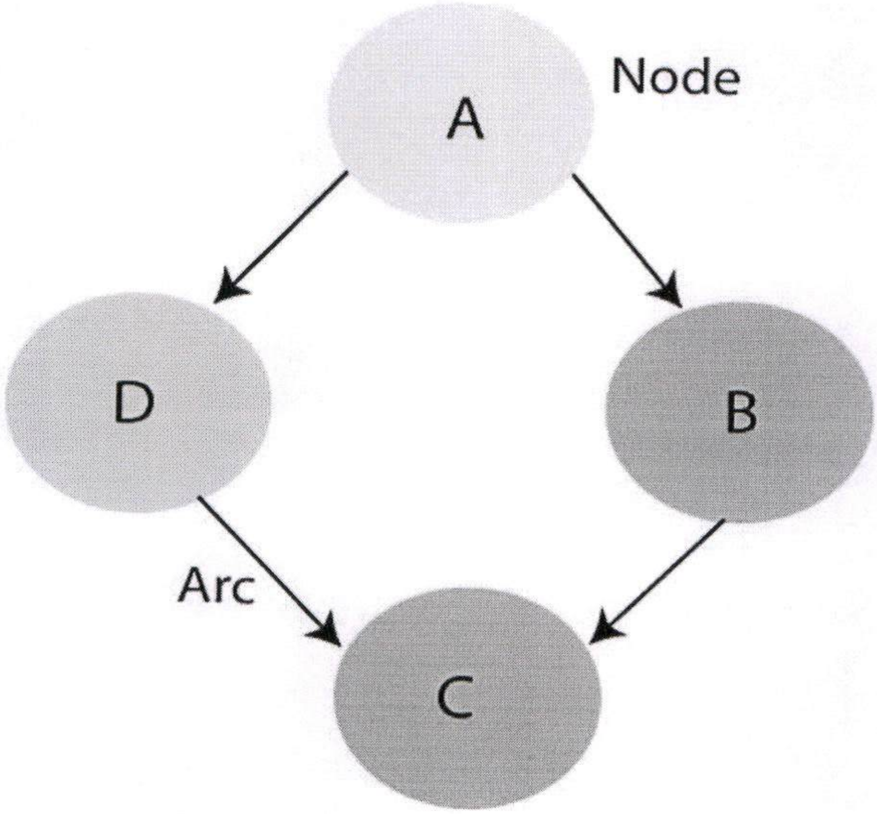
P(Face|King): probability of face card when we assume it is a king = 1

Putting all values in equation (i) we will get:

- **Directed Acyclic Graph**
- **Table of conditional probabilities.**

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**.

A Bayesian network graph is made up of nodes and Arcs (directed links), where:



- Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.
- **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph. These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other
    - **In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.**

$$P(king|face) = \frac{1 * (\frac{1}{13})}{(\frac{3}{13})}$$ **= 1/3, it is a probability that a face card is a king card.**

Application of Bayes' theorem in Artificial intelligence:

Following are some applications of Bayes' theorem:

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.
- It can solve the Monty Hall problem.

Bayesian Belief Network in artificial intelligence

Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty. We can define a Bayesian network as:

"A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."

It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.

Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction and anomaly detection.

Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction**, and **decision making under uncertainty**.

Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

- **If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.**
- **Node C is independent of node A.**

Note: The Bayesian network graph does not contain any cyclic graph. Hence, it is known as a **directed acyclic graph or DAG**.

The Bayesian network has mainly two components:

- **Causal Component**
- **Actual numbers**

Each node in the Bayesian network has condition probability distribution **$P(X_i | Parent(X_i))$**, which determines the effect of the parent on that node.

Bayesian network is based on Joint probability distribution and conditional probability. So let's first understand the joint probability distribution:

Joint probability distribution:

If we have variables x1, x2, x3,....., xn, then the probabilities of a different combination of x1, x2, x3.. xn, are known as Joint probability distribution.

$P[x_1, x_2, x_3,....., x_n]$, it can be written as the following way in terms of the joint probability distribution.

$= P[x_1 | x_2, x_3,....., x_n]P[x_2, x_3,....., x_n]$

$= P[x_1 | x_2, x_3,....., x_n]P[x_2|x_3,....., x_n]....P[x_{n-1}|x_n]P[x_n]$.

In general for each variable Xi, we can write the equation as:

$$P(X_i|X_{i-1},........., X_1) = P(X_i |Parents(X_i))$$

Explanation of Bayesian network:

Let's understand the Bayesian network through an example by creating a directed acyclic graph:

Example: Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry special case of resolution with a particular control strategy to decide which resolution to perform next.

UNIT-IV

Planning
Classical Planning: Definition of Classical Planning, Algorithms for Planning with StateSpace Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.
Planning and Acting in the Real World: Time, Schedules, and Resources, Hierarchical Planning, Planning and Acting in Nondeterministic Domains, Multi agent Planning

Planning Classical Planning: AI as the study of rational action, which means that planning—devising a plan of action to achieve one's goals—is a critical part of AI. We have seen two examples of planning agents so far the search-based problem-solving agent.

DEFINITION OF CLASSICAL PLANNING: The problem-solving agent can find sequences of actions that result in a goal state. But it deals with atomic representations of states and thus needs good domain-specific heuristics to perform well. The hybrid propositional logical agent can find plans without domain-specific heuristics because it uses domain-independent heuristics based on the logical structure of the problem but it relies on ground (variable-free) propositional inference,whichmeansthatitmaybeswampedwhentherearemanyactionsandstates.For example,intheworld,thesimpleactionofmovingastepforwardhadtoberepeated for all four agent orientations, T time steps, and n^2 currentlocations.

In response to this, planning researchers have settled on a **factored representation**— one in which a state of the world is represented by a collection of variables. We use a language called **PDDL**, the Planning Domain Definition Language that allows us to express all $4Tn^2$ actions with one action schema. There have been several versions of PDDL.we select a simple version and alter its syntax to be consistent with the rest of the book. We now show how PDDL describes the four things we need to define a search problem: the initial state, the actions that are available in a state, the result of applying an action, and the goal test.

Each state is represented as a conjunction of flaunts that are ground, functionless atoms. For example, Poor ∧ Unknown might represent the state of a hapless agent, and a state in a package delivery problem might be At(Truck 1, Melbourne) ∧ At(Truck 2, Sydney). Database semantics is used: the closed-world assumption means that any flaunts that are not mentioned are false, and the unique names assumption means that Truck 1 and Truck 2 are distinct.

A set of ground (variable-free) actions can be represented by a single action schema. The schema is a lifted representation—it lifts the level of reasoning from propositional logic to a restricted subset of first-order logic. For example, here is an action schema for flying a plane from one location to another:

Action(Fly (p, from, to),

PRECOND:At(p, from) ∧ Plane(p) ∧ Airport (from) ∧ Airport (to)

EFFECT:¬At(p, from) ∧ At(p, to))

The schema consists of the action name, a list of all the variables used in the schema, a precondition and an effect.

A set of action schemas serves as a definition of a planning domain. A specific problem within the domain is defined with the addition of an initial state and a goal.

state is a conjunction of ground atoms. (As with all states, the closed-world assumption is used, which means that any atoms that are not mentioned are false.) The goal is just like a precondition: a conjunction of literals (positive or negative) that may contain variables, such as At(p, SFO) ∧ Plane(p). Any variables are treated as existentially quantified, so this goal is to have any plane at SFO. The problem is solved when we can find a sequence of actions that end in a states that entails the goal.

Example: Air cargo transport

An air cargo transport problem involving loading and unloading cargo and flying it from place to place. The problem can be defined with three actions: Load , Unload , and Fly . The actions affect two predicates: In(c, p) means that cargo c is inside plane p, and At(x, a) means that object x (either plane or cargo) is at airport a. Note that some care must be taken to make sure the At predicates are maintained properly. When a plane flies from one airport to another, all the cargo inside the plane goes with it. In first-order logic it would be easy to quantify over all objects that are inside the plane. But basic PDDL does not have a universal quantifier, so we need a different solution. The approach we use is to say that a piece of cargo ceases to be At anywhere when it is In a plane; the cargo only becomes At the new airport when it is unloaded. So At really means "available for use at a given location."

The complexity of classical planning :

We consider the theoretical complexity of planning and distinguish two decision problems. PlanSAT is the question of whether there exists any plan that solves a planning problem. Bounded PlanSAT asks whether there is a solution of length k or less; this can be used to find an optimal plan.

The first result is that both decision problems are decidable for classical planning. The proof follows from the fact that the number of states is finite. But if we add function symbols to the language, then the number of states becomes infinite, and PlanSAT becomes only semi decidable: an algorithm exists that will terminate with the correct answer for any solvable problem, but may not terminate on unsolvable problems. The Bounded PlanSAT problem remains decidable even in the presence of function symbols.

Both PlanSAT and Bounded PlanSAT are in the complexity class PSPACE, a class that is larger (and hence more difficult) than NP and refers to problems that can be solved by a deterministic Turing machine with a polynomial amount of space. Even if we make some rather severe restrictions, the problems remain quite difficult.

Algorithms for Planning with State Space Search

Forward (progression) state-space search:

Now that we have shown how a planning problem maps into a search problem, we can solve planning problems with any of the heuristic search algorithms from Chapter 3 or a local search algorithm from Chapter 4 (provided we keep track of the actions used to reach the goal). From the earliest days of planning research (around 1961) until around 1998 it was assumed that forward state-space search was too inefficient to be practical. It is not hard to come up with reasons why .

First, forward search is prone to exploring irrelevant actions. Consider the noble task of buying a copy of AI: A Modern Approach from an online bookseller. Suppose there is an action schema Buy(isbn) with effect Own(isbn). ISBNs are 10 digits, so this action schema represents 10 billion ground actions. An uninformed forward-search algorithm would have to start enumerating these 10 billion actions to find one that leads to the goal.

Second, planning problems often have large state spaces. Consider an air cargo problem with 10 airports, where each airport has 5 planes and 20 pieces of cargo. The goal is to move all the cargo at airport A to airport B. There is a simple solution to the problem: load the 20 pieces of cargo into one of the planes at A, fly the plane to B, and unload the cargo. Finding the solution can be difficult because the average branching factor is huge: each of the 50 planes can fly to 9 other airports, and each of the 200 packages can be either unloaded (if it is loaded) or loaded into any plane at its airport (if it is unloaded). So in any state there is a minimum of 450 actions (when all the packages are at airports with no planes) and a maximum of 10,450 (when all packages and planes are at the same airport). On average, let's say there are about 2000 possible actions per state, so the search graph up to the depth of the obvious solution has about 2000 nodes.

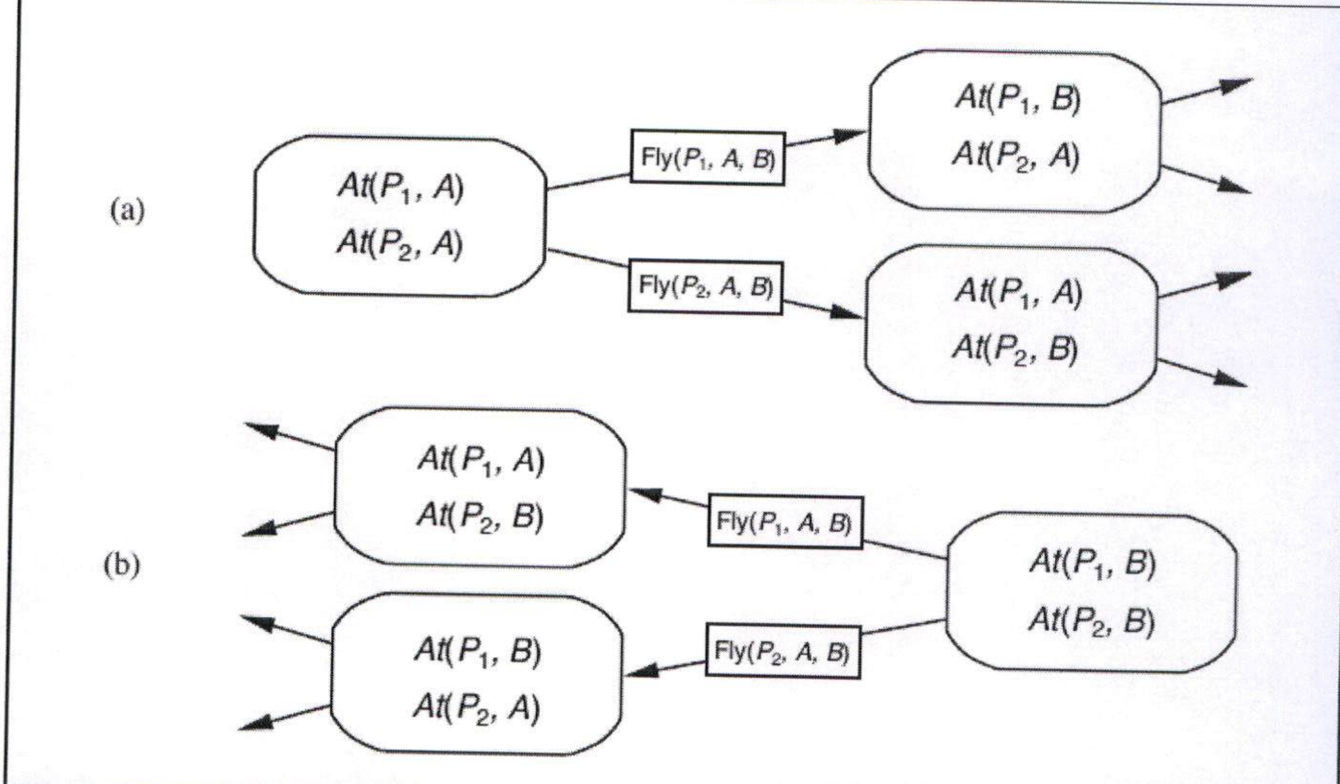


Figure 10.5 Two approaches to searching for a plan. (a) Forward (progression) search through the space of states, starting in the initial state and using the problem's actions to search forward for a member of the set of goal states. (b) Backward (regression) search through sets of relevant states, starting at the set of states representing the goal and using the inverse of the actions to search backward for the initial state.

Backward (regression) relevant-states search:

In regression search we start at the goal and apply the actions backward until we find a sequence of steps that reaches the initial state. It is called relevant-states search because we only consider actions that are relevant to the goal (or current state). As in belief-state search (Section 4.4), there is a set of relevant states to consider at each step, not just a single state.

We start with the goal, which is a conjunction of literals forming a description of a set of states—for example, the goal ¬Poor ∧ Famous describes those states in which Poor is false, Famous is true, and any other fluent can have any value. If there are n ground flaunts in a domain, then there are 2n ground states (each fluent can be true or false), but 3n descriptions of sets of goal states (each fluent can be positive, negative, or not mentioned).

In general, backward search works only when we know how to regress from a state description to the predecessor state description. For example, it is hard to search backwards for a solution to the n-queens problem because there is no easy way to describe the states that are one move away from the goal. Happily, the PDDL representation was designed to make it easy to regress actions—if a domain can be expressed in PDDL, then we can do regression search on it.

The final issue is deciding which actions are candidates to regress over. In the forward direction we chose actions that were applicable—those actions that could be the next step in the plan. In backward search we want actions that are relevant—those actions that could be the last step in a plan leading up to the current goal state.

Heuristics for planning:

Neither forward nor backward search is efficient without a good heuristic function. Recall from Chapter 3 that a heuristic function h(s) estimates the distance from a state s to the goal and that if we can derive an admissible heuristic for this distance—one that does not overestimate—then we can use A∗ search to find optimal solutions. An admissible heuristic can be derived by defining a relaxed problem that is easier to solve. The exact cost of a solution to this easier problem then becomes the heuristic for the original problem.

By definition, there is no way to analyze an atomic state, and thus it it requires some ingenuity by a human analyst to define good domain-specific heuristics for search problems with atomic states. Planning uses a factored representation for states and action schemas. That makes it possible to define good domain-independent heuristics and for programs to automatically apply a good domain-independent heuristic for a given problem.

Planning Graphs:

All of the heuristics we have suggested can suffer from inaccuracies. This section shows how a special data structure called a planning graph can be used to give better heuristic estimates. These heuristics can be applied to any of the search techniques we have seen so far. Alternatively, we can search for a solution over the space formed by the planning graph, using an algorithm called GRAPHPLAN.

A planning problem asks if we can reach a goal state from the initial state. Suppose we are given a tree of all possible actions from the initial state to successor states, and their successors, and so on. If we indexed this tree appropriately, we could answer the planning question "can we reach state G from state S0" immediately, just by looking it up. Of course, the tree is of exponential size, so this approach is impractical. A planning graph is polynomial- size approximation to this tree that can be constructed quickly. The planning graph can't answer definitively whether G is reachable from S0, but it can estimate how many steps it takes to reach G. The estimate is always correct when it reports the goal is not reachable, and it never overestimates the number of steps, so it is an admissible heuristic.

A planning graph is a directed graph organized into levels: first a level S0 for the initial state, consisting of nodes representing each fluent that holds in S0; then a level A0 consisting of nodes for each ground action that might be applicable in S0; then alternating levels Si followed by Ai; until we reach a termination condition (to be discussed later).

Roughly speaking, Si contains all the literals that could hold at time i, depending on the actions executed at preceding time steps. If it is possible that either P or ¬P could hold, then both will be represented in Si. Also roughly speaking, Ai contains all the actions that could have their preconditions satisfied at time i. We say "roughly speaking" because the planning graph records only a restricted subset of the possible negative interactions among actions; therefore, a literal might show up at level Sj when actually it could not be true until a later level, if at all. (A literal will never show up too late.) Despite the possible error, the level j at which a literal first appears is a good estimate of how difficult it is to achieve the literal from the initial state.

We now define mutex links for both actions and literals. A mutex relation holds between two actions at a given level if any of the following three conditions holds:

- Inconsistent effects: one action negates an effect of the other. For example, Eat(Cake) and the persistence of Have(Cake) have inconsistent effects because they disagree on the effect Have(Cake).
- Interference: one of the effects of one action is the negation of a precondition of the other. For example Eat(Cake) interferes with the persistence of Have(Cake) by its precondition.
- Competing needs: one of the preconditions of one action is mutually exclusive with a precondition of the other. For example, Bake(Cake) and Eat(Cake) are mutex because they compete on the value of the Have(Cake) precondition.

A mutex relation holds between two literals at the same level if one is the negation of the other or if each possible pair of actions that could achieve the two literals is mutually exclusive. This condition is called inconsistent support. For example, Have(Cake) and Eaten(Cake) are mutex in S1 because the only way of achieving Have(Cake), the persistence action, is mutex with the only way of achieving Eaten(Cake), namely Eat(Cake). In S2 the two literals are not mutex, because there are new ways of achieving them, such as Bake(Cake) and the persistence of Eaten(Cake), that are not mutex.

other Classical Planning Approaches:

Currently the most popular and effective approaches to fully automated planning are:

- Translating to a Boolean satisfiability (SAT) problem
- Forward state-space search with carefully crafted heuristics
- Search using a planning graph (Section 10.3)

These three approaches are not the only ones tried in the 40-year history of automated planning. Figure 10.11 shows some of the top systems in the International Planning Competitions, which have been held every even year since 1998. In this section we first describe the translation to a satisfiability problem and then describe three other influential approaches: planning as first-order logical deduction; as constraint satisfaction; and as plan refinement.

Classical planning as Boolean satisfiability :

we saw how SATPLAN solves planning problems that are expressed in propositional logic. Here we show how to translate a PDDL description into a form that can be processed by SATPLAN. The translation is a series of straightforward steps:

- Proposition Alize the actions: replace each action schema with a set of ground actions formed by substituting constants for each of the variables. These ground actions are not part of the translation, but will be used in subsequent steps.
- Define the initial state: assert F 0 for every fluent F in the problem's initial state, and ¬F for every fluent not mentioned in the initial state.
- Proposition Alize the goal: for every variable in the goal, replace the literals that contain the variable with a disjunction over constants. For example, the goal of having block A on another block, On(A, x) ∧ Block (x) in a world with objects A, B and C, would be replaced by the goal

(On(A, A) ∧ Block (A)) ∨ (On(A, B) ∧ Block (B)) ∨ (On(A, C) ∧ Block (C)) .

- Add successor-state axioms: For each fluent F , add an axiom of the form

F t+1 ⇔ ActionCausesF t ∨ (F t ∧ ¬ActionCausesNotF t) ,

where Action CausesF is a disjunction of all the ground actions that have F in their add list, and Action CausesNotF is a disjunction of all the ground actions that have F in their delete list.

Analysis of Planning approaches:

Planning combines the two major areas of AI we have covered so far: search and logic. A planner can be seen either as a program that searches for a solution or as one that (constructively) proves the existence of a solution. The cross-fertilization of ideas from the two areas has led both to improvements in performance amounting to several orders of magnitude in the last decade and to an increased use of planners in industrial applications. Unfortunately, we do not yet have a clear understanding of which techniques work best on which kinds of problems. Quite possibly, new techniques will emerge that dominate existing methods.

Planning is foremost an exercise in controlling combinatorial explosion. If there are n propositions in a domain, then there are 2n states. As we have seen, planning is PSPACE- hard. Against such pessimism, the identification of independent sub problems can be a powerful weapon. In the best case—full decomposability of the problem—we get an exponential speedup.

Decomposability is destroyed, however, by negative interactions between actions. GRAPHPLAN records mutexes to point out where the difficult interactions are. SATPLAN rep- resents a similar range of mutex relations, but does so by using the general CNF form rather than a specific data structure. Forward search addresses the problem heuristically by trying to find patterns (subsets of propositions) that cover the independent sub problems. Since this approach is heuristic, it can work even when the sub problems are not completely independent.

Sometimes it is possible to solve a problem efficiently by recognizing that negative interactions can be ruled out. We say that a problem has serializable sub goals if there exists an order of sub goals such that the planner can achieve them in that order without having to undo any of the previously achieved sub goals. For example, in the blocks world, if the goal is to build a tower (e.g., A on B, which in turn is on C, which in turn is on the Table, as in Figure 10.4 on page 371), then the sub goals are serializable bottom to top: if we first achieve C on Table, we will never have to undo it while we are achieving the other sub goals. Planners such as GRAPHPLAN, SATPLAN, and FF have moved the field of planning forward, by raising the level of performance of planning systems.

Planning and Acting in the Real World:

This allows human experts to communicate to the planner what they know about how to solve the problem. Hierarchy also lends itself to efficient plan construction because the planner can solve a problem at an abstract level before delving into details. Presents agent architectures that can handle uncertain environments and interleave deliberation with execution, and gives some examples of real-world systems.

Time, Schedules, and Resources:

The classical planning representation talks about what to do, and in what order, but the representation cannot talk about time: how long an action takes and when it occurs. For example, the planners of Chapter 10 could produce a schedule for an airline that says which planes are assigned to which flights, but we really need to know departure and arrival times as well. This is the subject matter of scheduling. The real world also imposes many resource constraints; for example, an airline has a limited number of staff—and staff who are on one flight cannot be on another at the same time. This section covers methods for representing and solving planning problems that include temporal and resource constraints.

The approach we take in this section is "plan first, schedule later": that is, we divide the overall problem into a planning phase in which actions are selected, with some ordering constraints, to meet the goals of the problem, and a later scheduling phase, in which temporal information is added to the plan to ensure that it meets resource and deadline constraints.

$Jobs(\{AddEngine1 \prec AddWheels1 \prec Inspect1\},$
$\quad \{AddEngine2 \prec AddWheels2 \prec Inspect2\})$

$Resources(EngineHoists(1), WheelStations(1), Inspectors(2), LugNuts(500))$

$Action(AddEngine1, \text{DURATION}:30,$
$\quad \text{USE}: EngineHoists(1))$
$Action(AddEngine2, \text{DURATION}:60,$
$\quad \text{USE}: EngineHoists(1))$
$Action(AddWheels1, \text{DURATION}:30,$
$\quad \text{CONSUME}: LugNuts(20), \text{USE}: WheelStations(1))$
$Action(AddWheels2, \text{DURATION}:15,$
$\quad \text{CONSUME}: LugNuts(20), \text{USE}: WheelStations(1))$
$Action(Inspect_i, \text{DURATION}:10,$
$\quad \text{USE}: Inspectors(1))$

Figure 11.1 A job-shop scheduling problem for assembling two cars, with resource constraints. The notation $A \prec B$ means that action A must precede action B.

This approach is common in real-world manufacturing and logistical settings, where the planning phase is often performed by human experts. The automated methods of Chapter 10 can also be used for the planning phase, provided that they produce plans with just the minimal ordering constraints required for correctness. G RAPHPLAN (Section 10.3), SATPLAN (Section 10.4.1), and partial-order planners (Section 10.4.4) can do this; search-based methods (Section 10.2) produce totally ordered plans, but these can easily be converted to plans with minimal ordering constraints.

Hierarchical Planning :

The problem-solving and planning methods of the preceding chapters all operate with a fixed set of atomic actions. Actions can be strung together into sequences or branching networks; state-of-the-art algorithms can generate solutions containing thousands of actions.

For plans executed by the human brain, atomic actions are muscle activations. In very round numbers, we have about 103 muscles to activate (639, by some counts, but many of them have multiple subunits); we can modulate their activation perhaps 10 times per second; and we are alive and awake for about 109 seconds in all. Thus, a human life contains about 1013 actions, give or take one or two orders of magnitude. Even if we restrict ourselves to planning over much shorter time horizons—for example, a two-week vacation in Hawaii—a detailed motor plan would contain around 1010 actions. This is a lot more than 1000.

To bridge this gap, AI systems will probably have to do what humans appear to do: plan at higher levels of abstraction. A reasonable plan for the Hawaii vacation might be "Go to San Francisco airport; take Hawaiian Airlines flight 11 to Honolulu; do vacation stuff for two weeks; take Hawaiian Airlines flight 12 back to San Francisco; go home." Given such a plan, the action "Go to San Francisco airport" can be viewed as a planning task in itself, with a solution such as "Drive to the long-term parking lot; park; take the shuttle to the terminal." Each of these actions, in turn, can be decomposed further, until we reach the level of actions that can be executed without deliberation to generate the required motor control sequence.

Planning and Acting in Nondeterministic Domains: While the basic concepts are the same as in Chapter 4, there are also significant differences. These arise because planners deal with factored representations rather than atomic representations. This affects the way we represent the agent's capability for action and observation and the way we represent belief states—the sets of possible physical states the agent might be in—for unobservable and partially observable environments. We can also take ad- vantage of many of the domain-independent methods given in Chapter 10 for calculating search heuristics.

Consider this problem: given a chair and a table, the goal is to have them match—have the same color. In the initial state we have two cans of paint, but the colors of the paint and the furniture are unknown. Only the table is initially in the agent's field of view:

Init(Object(Table) ∧ Object(Chair) ∧ Can(C1) ∧ Can(C2) ∧ InView (Table)) Goal (Color (Chair , c) ∧ Color (Table, c))

There are two actions: removing the lid from a paint can and painting an object using the paint from an open can. The action schemas are straightforward, with one exception: we now allow preconditions and effects to contain variables that are not part of the action's variable list. That is, Paint(x, can) does not mention the variable c, representing the color of the paint in the can. In the fully observable case, this is not allowed—we would have to name the action Paint(x, can, c). But in the partially observable case, we might or might not know what color is in the can. (The variable c is universally quantified, just like all the other variables in an action schema.)

Action(RemoveLid (can),

PRECOND:Can(can)

EFFECT:Open(can))

Action(Paint(x , can),

PRECOND:Object(x) $\wedge$ Can(can) $\wedge$ Color (can, c) $\wedge$ Open(can)

EFFECT:Color (x, c))

To solve a partially observable problem, the agent will have to reason about the percepts it will obtain when it is executing the plan. The percept will be supplied by the agent's sensors when it is actually acting, but when it is planning it will need a model of its sensors. In Chapter 4, this model was given by a function, PERCEPT(s). For planning, we augment PDDL with a new type of schema, the percept schema:

Multi agent Planning:

we have assumed that only one agent is doing the sensing, planning, and acting. When there are multiple agents in the environment, each agent faces a multi agent planning problem in which it tries to achieve its own goals with the help or hindrance of others.

Between the purely single-agent and truly multi agent cases is a wide spectrum of problems that exhibit various degrees of decomposition of the monolithic agent. An agent with multiple effectors that can operate concurrently—for example, a human who can type and speak at the same time—needs to do multi effector planning to manage each effector while handling positive and negative interactions among the effectors. When the effectors are physically decoupled into detached units—as in a fleet of delivery robots in a factory— multi effector planning becomes multibody planning. A multibody problem is still a "standard" single-agent problem as long as the relevant sensor information collected by each body can be pooled—either centrally or within each body—to form a common estimate of the world state that then informs the execution of the overall plan; in this case, the multiple bodies act as a single body.

When a single entity is doing the planning, there is really only one goal, which all the bodies necessarily share. When the bodies are distinct agents that do their own planning, they may still share identical goals; for example, two human tennis players who form a doubles team share the goal of winning the match. Even with shared goals, however, the multibody and multi agent cases are quite different. In a multibody robotic doubles team, a single plan dictates which body will go where on the court and which body will hit the ball. In a multi- agent doubles team, on the other hand, each agent decides what to do; without some method for coordination, both agents may decide to cover the same part of the court and each may leave the ball for the other to hit.

Planning with multiple simultaneous actions

UNIT-V

Uncertainty: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use, Probabilistic Reasoning: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.
Learning: Forms of Learning, Supervised Learning, Learning Decision Trees.Knowledge in Learning: Logical Formulation of Learning, Knowledge in Learning, Explanation-Based Learning, Learning Using Relevance Information, Inductive Logic Programming

Uncertain knowledge and Learning

Core vs. Probabilistic AI •

- Knowledge Reasoning : work with facts/assertions; develop rules of logical inference
- Planning: work with applicability/effects of actions; develop searches for actions which achieve goals/avert disasters.
- Expert Systems: develop by hand a set of rules for examining inputs, updating internal states and generating outputs
- Learning approach: use probabilistic models to tune performance based on many data examples.
- Probabilistic AI: emphasis on noisy measurements, approximation in hard cases, learning, algorithmic issues.
    - logical assertions ⇒ probability distributions
    - logical inference ⇒ conditional probability distributions
    - logical operators ⇒ probabilistic generative models

Probabilistic reasoning

Causes of uncertainty:

Following are some leading causes of uncertainty to occur in the real world.

- Information occurred from unreliable sources.
- Experimental Errors
- Equipment fault
- Temperature variation
- Climate change.

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.

We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of someone's laziness and ignorance.

In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today," "behavior of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

Probability: Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

$0 \leq P(A) \leq 1$, where P(A) is the probability of an event A.

P(A) = 0, indicates total uncertainty in an event A.

P(A) =1, indicates total certainty in an event A.

We can find the probability of an uncertain event by using the below formula.
P(¬A) = probability of a not happening event.
P(¬A) + P(A) = 1.

Event: Each possible outcome of a variable is called an event.

Sample space: The collection of all possible events is called sample space.

Random variables: Random variables are used to represent the events and objects in the real world.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior Probability: The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

Conditional probability:

Conditional probability is a probability of occurring an event when another event has already happened.

Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

Where P(A⋀B)= Joint probability of a and B

P(B)= Marginal probability of B.

If the probability of A is given and we need to find the probability of B, then it will be given as:

It can be explained by using the below Venn diagram, where B is occurred event, so sample space will be reduced to set B, and now we can only calculate event A when event B is already occurred by dividing the probability of P(A⋀B) by P(B).

Example:

In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like English also like mathematics?

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

Hence, 57% are the students who like English also like Mathematics.

Why Reason Probabilistically?

- In many problem domains it isn't possible to create complete, consistent models of the world. Therefore agents (and people) must act in uncertain worlds (which the real world is).
- Want an agent to make rational decisions even when there is not enough information to prove that an action will work.
- Some of the reasons for reasoning under uncertainty:
  - **True uncertainty**. E.g., flipping a coin.
  - **Theoretical ignorance**. There is no complete theory which is known about the problem domain. E.g., medical diagnosis.
  - **Laziness**. The space of relevant factors is very large, and would require too much work to list the complete set of antecedents and consequents. Furthermore, it would be too hard to use the enormous rules that resulted.
  - **Practical ignorance**. Uncertain about a particular individual in the domain because all of the information necessary for that individual has not been collected.

- Probability theory will serve as the formal language for representing and reasoning with uncertain knowledge.

Bayes' theorem:

Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.

Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics

It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).

Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Example: If cancer corresponds to one's age then by using Bayes' theorem, we can determine the probability of cancer more accurately with the help of age.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:

As from product rule we can write:

1. $P(A \wedge B) = P(A|B) P(B)$ or

Similarly, the probability of event B with known event A:

1. $P(A \wedge B) = P(B|A) P(A)$

Equating right hand side of both the equations, we will get:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad \text{....(a)}$$

The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference.**

It shows the simple relationship between joint and conditional probabilities. Here,

P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

P(B|A) is called the likelihood, in which we consider that hypothesis is true, then we calculate the probability of evidence.

P(A) is called the **prior probability**, probability of hypothesis before considering the evidence

P(B) is called **marginal probability**, pure probability of an evidence.

In the equation (a), in general, we can write P (B) = P(A)*P(B|Ai), hence the Bayes' rule can be written as:

$$P(A_i|B) = \frac{P(A_i)*P(B|A_i)}{\sum_{i=1}^{k} P(A_i)*P(B|A_i)}$$

Applying Bayes' rule:

Bayes' rule allows us to compute the single term P(B|A) in terms of P(A|B), P(*B*), and P(A). This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one.

Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

$$P(cause|effect) = \frac{P(effect|cause)\ P(cause)}{P(effect)}$$

Example-1:

Question: what is the probability that a patient has diseases meningitis with a stiff neck?

Given Data:

A doctor is aware that disease meningitis causes a patient to have a stiff neck, and it occurs 80% of the time. He is also aware of some more facts, which are given as follows:

- The Known probability that a patient has meningitis disease is 1/30,000.
- The Known probability that a patient has a stiff neck is 2%.

Let a be the proposition that patient has stiff neck and b be the proposition that patient has meningitis. , so we can calculate the following as:

P(a|b) = 0.8

P(b) = 1/30000

P(a)= .02

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)} = \frac{0.8*(\frac{1}{30000})}{0.02} = 0.001333333.$$

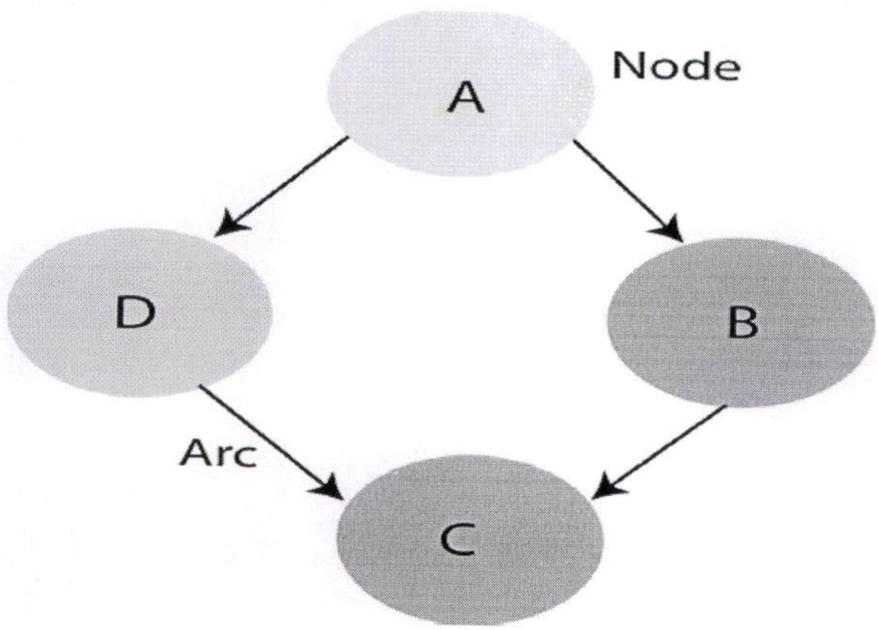
Hence, we can assume that 1 patient out of 750 patients has meningitis disease with a stiff neck.

Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

- **Directed Acyclic Graph**
- **Table of conditional probabilities.**

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram.**

A Bayesian network graph is made up of nodes and Arcs (directed links), where:



- Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.
- **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph. These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other
  - **In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.**
  - **If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.**
  - **Node C is independent of node A.**

The Bayesian network has mainly two components:

- **Causal Component**
- **Actual numbers**

Each node in the Bayesian network has condition probability distribution **$P(X_i \mid Parent(X_i))$**, which determines the effect of the parent on that node.

Representing Belief about Propositions

- Rather than reasoning about the truth or falsity of a proposition, reason about the belief that a proposition or event is true or false
- For each primitive proposition or event, attach a **degree of belief** to the sentence
- Use **probability theory** as a formal means of manipulating degrees of belief
- Given a proposition, A, assign a probability, P(A), such that 0 <= P(A) <= 1, where if A is true, P(A)=1, and if A is false, P(A)=0. Proposition A must be either true or false, but P(A) summarizes our degree of belief in A being true/false.
  - Examples
  - P(Weather=Sunny) = 0.7 means that we believe that the weather will be Sunny with 70% certainty. In this case Weather is a random variable that can take on values in a domain such as {Sunny, Rainy, Snowy, Cloudy}.
  - P(Cavity=True) = 0.05 means that we believe there is a 5% chance that a person has a cavity. Cavity is a Boolean random variable since it can take on possible values *True* and *False*.
  - Example: P(A=a ^ B=b) = P(A=a, B=b) = 0.2, where A=My_Mood, a=happy, B=Weather, and b=rainy, means that there is a 20% chance that when it's raining my mood is happy.

- We will assume that in a given problem domain, the programmer and expert identify all of the relevant propositional variables that are needed to reason about the domain.
- Each of these will be represented as a **random variable**, i.e., a variable that can take on values from a set of mutually exclusive and exhaustive values called the **sample space** or **partition** of the random variable. Usually this will mean a sample space {*True*, *False*}.
- For example, the proposition *Cavity* has possible values *True* and *False* indicating whether a given patient has a cavity or not. A random variable that has True and False as its possible values is called a **Boolean random variable**.

More generally, propositions can include the equality predicate with random variables and the possible values they can have.

For example, we might have a random variable *Color* with possible values *red*, *green*, *blue*, and *other*.

Then P(Color=red) indicates the likelihood that the color of a given object is red.

Similarly, for Boolean random variables we can ask P(A=True), which is abbreviated to P(A), and P(A=False), which is abbreviated to P(~A).

Axioms of Probability Theory

Probability Theory provides us with the formal mechanisms and rules for manipulating propositions represented probabilistically. The following are the three axioms of probability theory:

- 0 <= P(A=a) <= 1 for all *a* in sample space of A
- P(True)=1, P(False)=0
- P(A v B) = P(A) + P(B) - P(A ^ B)

From these axioms we can show the following properties also hold:

- P(~A) = 1 - P(A)
- P(A) = P(A ^ B) + P(A ^ ~B)
- Sum{P(A=a)} = 1, where the sum is over all possible values *a* in the sample space of A

Joint Probability Distribution

Given an application domain in which we have determined a sufficient set of random variables to encode all of the relevant information about that domain, we can completely specify all of the possible probabilistic information by constructing the **full joint probability distribution,**
P(V1=v1, V2=v2, ..., Vn=vn), which assigns probabilities to all possible combinations of values to all random variables.

For example, consider a domain described by three Boolean random variables, Bird, Flier, and Young. Then we can enumerate a table showing all possible interpretations and associated probabilities:

Bird	Flier	Young	Probability
T	T	T	0.0
T	T	F	0.2
T	F	T	0.04
T	F	F	0.01
F	T	T	0.01
F	T	F	0.01
F	F	T	0.23
F	F	F	0.5

Notice that there are 8 rows in the above table representing the fact that there are 2^3 ways to assign values to the three Boolean variables. More generally, with *n* Boolean variables the table will be of size 2^n. And if *n* variables each had *k* possible values, then the table would be size k^n.

Also notice that the sum of the probabilities in the right column must equal 1 since we know that the set of all possible values for each variable are known. This means that for *n* Boolean random variables, the table has 2^n-1 values that must be determined to completely fill in the table.

If all of the probabilities are known for a full joint probability distribution table, then we can compute *any* probabilistic statement about the domain. For example, using the table above, we can compute

- P(Bird=T) = P(B) = 0.0 + 0.2 + 0.04 + 0.01 = 0.25
- P(Bird=T, Flier=F) = P(B, ~F) = P(B, ~F, Y) + F(B, ~F, ~Y) = 0.04 + 0.01 = 0.05

Conditional Probabilities

- Conditional probabilities are key for reasoning because they formalize the process of accumulating evidence and updating probabilities based on new evidence.
- For example, if we know there is a 4% chance of a person having a cavity, we can represent this as the **prior** (aka unconditional) probability P(Cavity)=0.04.
- Say that person now has a symptom of a toothache, we'd like to know what is the **posterior** probability of a Cavity given this new evidence. That is, compute P(Cavity | Toothache).
- If P(A|B) = 1, this is equivalent to the sentence in Propositional Logic B => A. Similarly, if P(A|B) =0.9, then this is like saying B => A with 90% certainty.
- In other words, we've made implication fuzzy because it's not absolutely certain.
- Given several measurements and other "evidence", E1, ..., Ek, we will formulate queries as P(Q | E1, E2, ..., Ek) meaning "what is the degree of belief that Q is true given that we know E1, ..., Ek *and nothing else*."

Conditional probability is defined as: P(A|B) = P(A ^ B)/P(B) = P(A,B)/P(B)
One way of looking at this definition is as a normalized (using P(B)) joint probability (P(A,B)).

- Example Computing Conditional Probability from the Joint Probability Distribution Say we want to compute P(~Bird | Flier) and we know the full joint probability distribution function given above.
- We can do this as follows:
- P(~B|F) = P(~B,F) / P(F)
- = (P(~B,F,Y) + P(~B,F,~Y)) / P(F)
- = (.01 + .01)/P(F)

Next, we could either compute the marginal probability P(F) from the full joint probability distribution, or, as is more commonly done, we could do it by using a process called **normalization**, which first requires computing

P(B|F) = P(B,F) / P(F)
= (P(B,F,Y) + P(B,F,~Y)) / P(F)
= (0.0 + 0.2)/P(F)

Now we also know that P(~B|F) + P(B|F) = 1, so substituting from above and solving for P(F) we get P(F) = 0.22. Hence, P(~B|F) = 0.02/0.22 = 0.091.

While this is an effective procedure for computing conditional probabilities, it is intractable in general because it means that we must compute and store the full joint probability distribution table, which is exponential in size.

- **Some important rules related to conditional probability are:**

- Rewriting the definition of conditional probability, we get the **Product Rule**: P(A,B) = P(A|B)P(B)
- **Chain Rule**: P(A,B,C,D) = P(A|B,C,D)P(B|C,D)P(C|D)P(D), which generalizes the product rule for a joint probability of an arbitrary number of variables. Note that ordering the variables results in a different expression, but all have the same resulting value.
- **Conditionalized version of the Chain Rule**: P(A,B|C) = P(A|B,C)P(B|C)
- **Bayes's Rule**: P(A|B) = (P(A)P(B|A))/P(B), which can be written as follows to more clearly emphasize the "updating" aspect of the rule: P(A|B) = P(A) * [P(B|A)/P(B)] Note: The terms P(A) and P(B) are called the **prior** (or **marginal**) probabilities. The term P(A|B) is called the **posterior** probability because it is derived from or depends on the value of B.
- **Conditionalized version of Bayes's Rule**: P(A|B,C) = P(B|A,C)P(A|C)/P(B|C)
- **Conditioning (aka Addition) Rule**: P(A) = Sum{P(A|B=b)P(B=b)} where the sum is over all possible values *b* in the sample space of B.
- P(~B|A) = 1 - P(B|A)

Assuming conditional independence of B and C given A, we can simplify Bayes's Rule for two pieces of evidence B and C:

- P(A | B,C) = (P(A)P(B,C | A))/P(B,C)
- = (P(A)P(B|A)P(C|A))/(P(B)P(C|B))
- = P(A) * [P(B|A)/P(B)] * [P(C|A)/P(C|B)]
- = (P(A) * P(B|A) * P(C|A))/P(B,C)

Naive Bayes Classifier:

Say we have a random variable, C, which represents the possible ways to classify an input pattern of features that have been measured.

The domain of C is the set of possible classifications, e.g., it might be the possible diagnoses in a medical domain.

Say the possible values for C are {a,b,c}, and the features we have measured are

E1=e1, E2=e2, ..., En=en.

Then we can compute

P(C=a | E1=e1, ..., En=en),

P(C=b | E1=e1, ..., En=en) and

P(C=c | E1=e1, ..., En=en) assuming E1, ..., En are conditionally independent given C.

Since for each value of C the denominators are the same above, they can be ignored.

So, for example

P(C=a | E1=e1, ..., En=en) = P(C=a) * P(E1=e1 | C=a) * P(E2=e2 | C=a) * ... * P(En=en | C=a)

Choose the value for C that gives the maximum probability.

Finally, since only relative values are needed and probabilities are often very small, it is common to compute the sum of logarithms of the probabilities:

log P(C=a | E1=e1, ..., En=en) = log P(C=a) + log P(E1=e1 | C=a) + ... + log P(En=en | C=a).

If B and C are (unconditionally) independent, then P(C|B) = P(C), so

P(A | B,C) = P(A) * [P(B|A)/P(B)] * [P(C|A)/P(C)]

- Example

Consider the medical domain consisting of three Boolean variables: PickledLiver, Jaundice, Bloodshot, where the first indicates if a given patient has the "disease" PickledLiver, and the second and third describe symptoms of the patient. We'll assume that Jaundice and Bloodshot are independent.

The doctor wants to determine the likelihood that the patient has a PickledLiver.

Based on no other information, she knows that the **prior** probability P(PickledLiver) = 10^{-17}. So, this represents the doctor's initial belief in this diagnosis. However, after examination, she determines that the patient has jaundice. She knows that P(Jaundice) = 2^{-10} and P(Jaundice | PickledLiver) = 2^{-3}, so she computes the new updated probability in the patient having PickledLiver as:

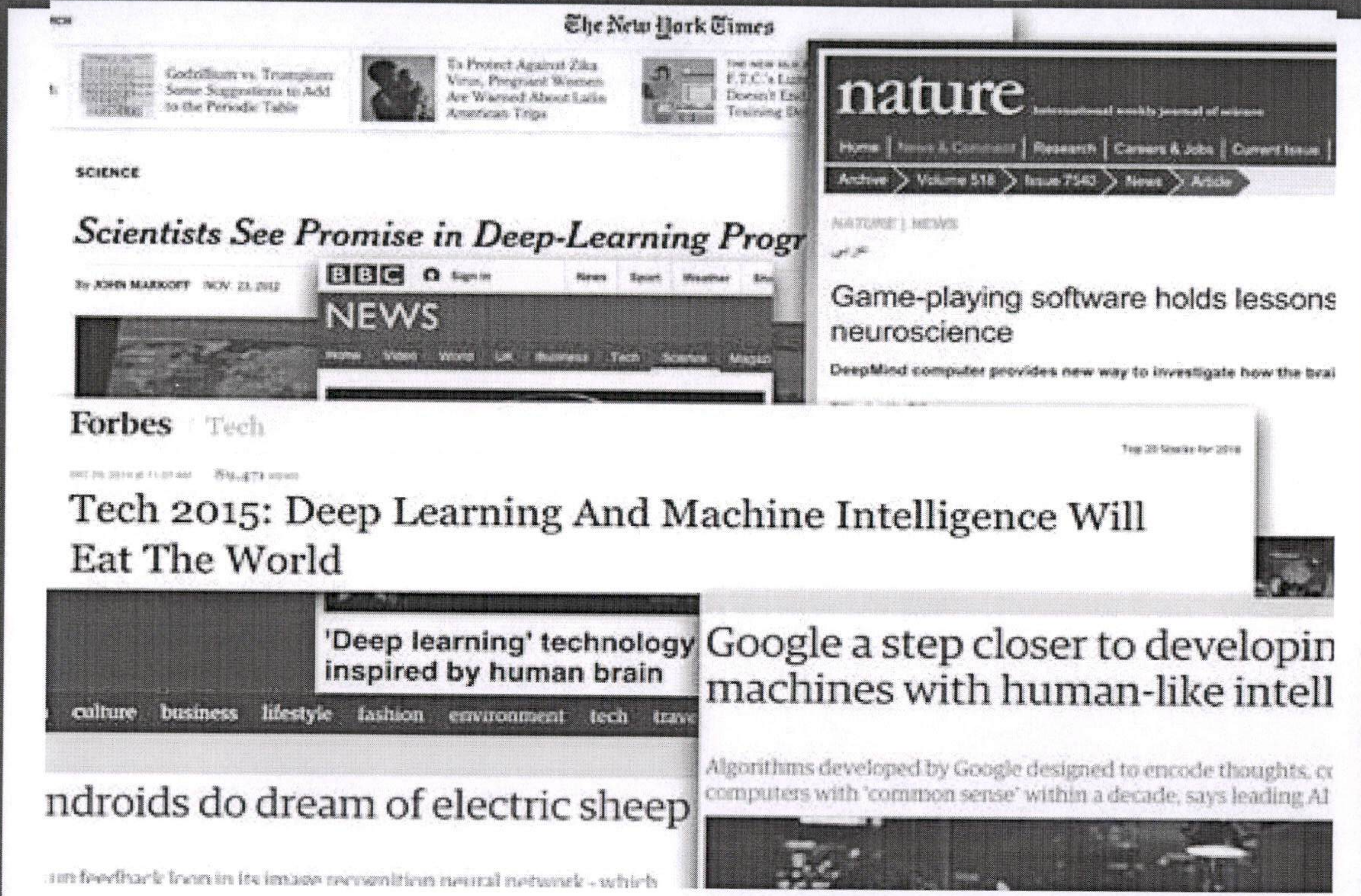
P(PickledLiver | Jaundice) = P(P)P(J|P)/P(J)
= $(2^{-17} * 2^{-3})/2^{-10}$
= 2^{-10}

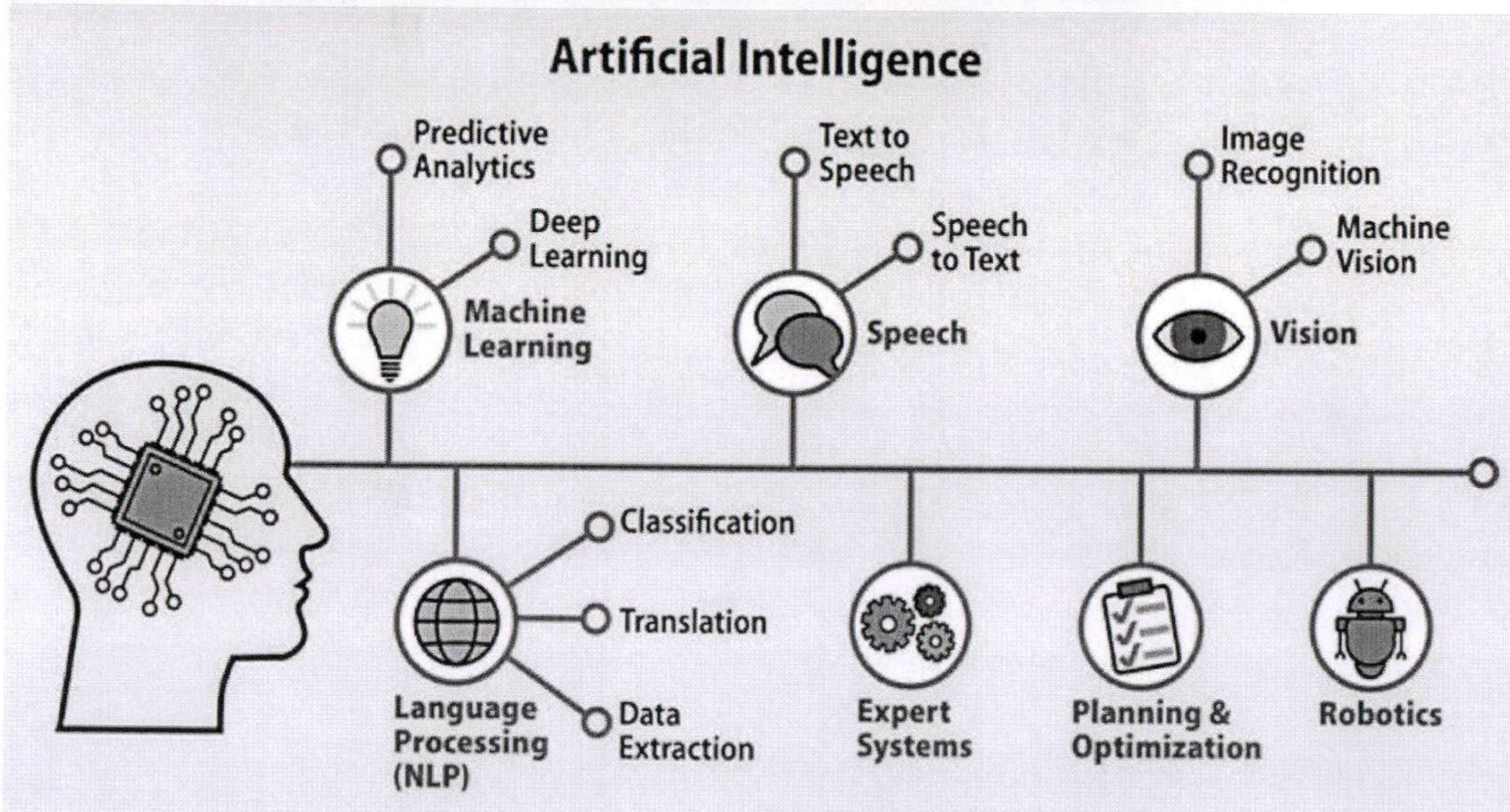
So, based on this new evidence, the doctor increases her belief in this diagnosis from 2^{-17} to 2^{-10}.

Next, she determines that the patient's eyes are bloodshot, so now we need to add this new piece of evidence and update the probability of PickledLiver given Jaundice and Bloodshot.

Say, P(Bloodshot) = 2^{-6} and P(Bloodshot | PickledLiver) = 2^{-1}. Then, she computes the new conditional probability:

Introduction to Artificial **Intelligence**


The New York Times
Godzillium vs. Trumpium: Some Suggestions to Add to the Periodic Table
To Protect Against Zika Virus, Pregnant Women Are Warned About Latin American Trips
SCIENCE
Scientists See Promise in Deep-Learning Progr
By JOHN MARKOFF NOV. 23, 2012
BBC
NEWS
nature
Home | News & Comment | Research | Careers & Jobs | Current Issue
Archive > Volume 518 > Issue 7540 > News > Article
NATURE | NEWS
Game-playing software holds lessons
neuroscience
DeepMind computer provides new way to investigate how the brai
Forbes Tech
Tech 2015: Deep Learning And Machine Intelligence Will Eat The World
'Deep learning' technology inspired by human brain
culture business lifestyle fashion environment tech
ndroids do dream of electric sheep
Google a step closer to developin
machines with human-like intell
Algorithms developed by Google designed to encode thoughts, c
computers with 'common sense' within a decade, says leading AI


Artificial Intelligence
Predictive Analytics
Deep Learning
Machine Learning
Text to Speech
Speech to Text
Speech
Image Recognition
Machine Vision
Vision
Classification
Translation
Data Extraction
Language Processing (NLP)
Expert Systems
Planning & Optimization
Robotics

History of A.I FROM 40's-90's

- 1946 : ENIAC was first electronic general purpose computers were developed.
- 1950 : Alan Turing asks the question **"CAN MACHINES THINK ?"**
- **1956 :** Dartmouth meeting: "Artificial Intelligence" adopted
- **Founders of A.I**
  - **J.McCarthy**
  - **Marvin Minsky**
  - **Alan Newell**
  - **Herbert Simon**

- 1950 **:** Turing Test for Machine Intelligence
- 1956 : AI born at Dartmouth Collage workshop
- 1964 : Eliza- Chatbot Psychotherapist
- 1966 : First general purpose mobile robot named Shakey was developed.

- 1974 -1980 : Winter (Failure of A.I)
- - Failure of machine translation
- - There were negative results in Neural nets.
- - Poor speech understanding

- Mid 90's: Cortes and Vapnik published paper on Support Vector Machines

- 2006: Hinton and Salakhutdinov proposed using Restricted Boltzmann Machine for pre-train Deep Neural Network
- 2007: Fei-Fei Li's ImageNet assembling a database of 14 million labelled images (Data drives learning)
- 2011: Microsoft explored Speech recognition and IBM's Watson
- 2014: Google acquired DeepMind, combing deep learning and reinforcement learning
- 2016: DeepMind's AlphaGo defeated world champion Lee Sedol
- Facebook: translate user posts in more than 40 languages
- Baidu: for speech recognition, translation, photo search and self-driving car project.

AI prehistory

- Philosophy — Logic, methods of reasoning, mind as physical system foundations of learning, language, rationality
- Mathematics — Formal representation and proof algorithms, computation, (un)decidability, (in)tractability, probability
- Economics — utility, decision theory
- Neuroscience — physical substrate for mental activity
- Psychology — phenomena of perception and motor control, experimental techniques
- Computer engineering — building fast computers
- Control theory — design systems that maximize an objective function over time
- Linguistics — knowledge representation, grammar

What is Artificial intelligence ?

Thinking Humanly "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)	**Thinking Rationally** "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)
Acting Humanly "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)	**Acting Rationally** "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)

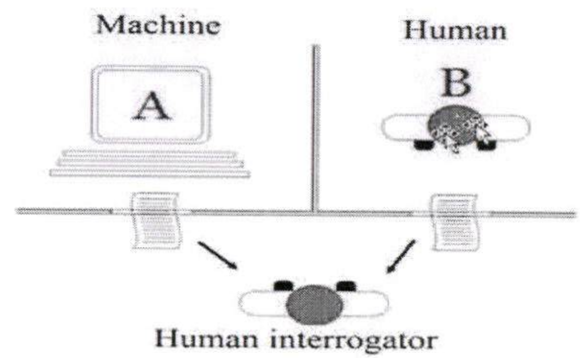
Figure 1.1 Some definitions of artificial intelligence, organized into four categories.

	Human-like Intelligence	"Ideal" Intelligent/ Rationally
Thought/ Reasoning	2.Thinking humanly	3.Thinking Rationally
Behavior/ Actions	1.Acting Humanly	4.Acting Rationally

Acting Humanly : Turing's Test

- If the human cannot tell whether the responses from the other side of a wall are coming from a human or computer, then computer is intelligent.

Turing test



- **Thinking humanly: The cognitive modeling approach**
- Requires scientific theories of internal activities of the brain.
- How to validate? Requires
  1)Predicting and testing behavior of human subjects (top- down)
  - 2) Direct identification from neurological data (bottom-up)
- Both approaches (roughly, Cognitive Science and Cognitive Neuroscience)

- **Thinking rationally: The "laws of thought" approach**
- The Greek philosopher Aristotle was one of the first to attempt to codify "right thinking," that is, irrefutable reasoning processes. – Logic
- 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: The rational agent approach**
- An agent is just something that acts. Of course, all computer programs do something, but computer agents are expected to do more: operate autonomously, perceive their environment, persist over a prolonged time period, adapt to change, and create and pursue goals.

- **An agent should strive to do the right thing, based on what it can perceive and the actions it can perform. A rational agent** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.
- **E.g., Performance measure of a vacuum cleaner agent could be amount of dust cleaned up, time taken , electricity consumed etc.**

1.1.4 Approaches to AI

Strong AI aims to build machines that can truly reason and solve problems. These machines should be self aware and their overall intellectual ability needs to be indistinguishable from that of a human being. Excessive optimism in the 1950s and 1960s concerning strong AI has given way to an appreciation of the extreme difficulty of the problem. Strong AI maintains that suitably programmed machines are capable of cognitive mental states.

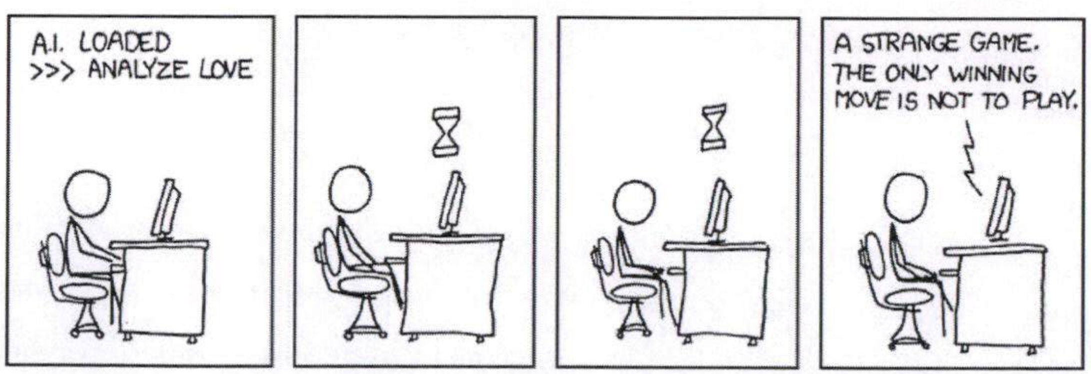
Weak AI: deals with the creation of some form of computer-based artificial intelligence that cannot truly reason and solve problems, but can act as if it were intelligent. Weak AI holds that suitably programmed machines can simulate human cognition.

Applied AI: aims to produce commercially viable "smart" systems such as, for example, a security system that is able to recognise the faces of people who are permitted to enter a particular building. Applied AI has already enjoyed considerable success.

Cognitive AI: computers are used to test theories about how the human mind works--for example, theories about how we recognise faces and other objects, or about how we solve abstract problems.

UNIT -II

Adversarial Search and Game Playing



http://xkcd.com/601/

Games

- Games: multi-agent environment
  - What do other agents do and how do they affect our success?
  - Cooperative vs. competitive multi-agent environments.
  - Competitive multi-agent environments give rise to adversarial search a.k.a. games
- Why study games?
  - Fun!
  - They are hard
  - Easy to represent and agents restricted to small number of actions… sometimes!

Relation of Games to Search

- Search – no adversary
  - Solution is (heuristic) method for finding goal
  - Heuristics and CSP techniques can find optimal solution
  - Evaluation function: estimate of cost from start to goal through given node
  - Examples: path planning, scheduling activities
- Games – adversary
  - Solution is strategy (strategy specifies move for every possible opponent reply).
  - Time limits force approximate solutions
  - Examples: chess, checkers, Othello, backgammon

Types of Games

•**Perfect information:** A game with the perfect information is that in which agents can look into the complete board. Agents have all the information about the game, and they can see each other moves also.

Examples Chess, Checkers, Go, etc.

•**Imperfect information:** If in a game agents do not have all information about the game and not aware with what's going on, such type of games are called the game with imperfect information

Ex: tic-tac-toe, Battleship, blind, Bridge, etc.

•**Deterministic games:** Deterministic games are those games which follow a strict pattern and set of rules for the games, and there is no randomness associated with them.

Examples : chess, Checkers, Go, tic-tac-toe, etc.

•**Non-deterministic games:** Non-deterministic are those games which have various unpredictable events and has a factor of chance or luck. This factor of chance or luck is introduced by either dice or cards. These are random, and each action response is not fixed.

Example: Backgammon, Monopoly, Poker, etc.

UNIT - III

First –Order Logic

- Propositional logic assumes the world contains facts.
- Propositional logic has limited expressive power.
  - E.g., : We can't say "pit causes breeze in adjacent squares", except by writing one sentence for each square.
  - First order logic is also known as Predicate logic or First order predicate logic.
  - FOL has well defined syntax and semantics.
  - FOL (like natural language) assumes the world contains :
  - **Objects** : People, colors ,houses etc.
  - **Relations** : brother of, bigger than, part of, comes between.

  If only 1 object then we call as unary relation.
  - **Functions** : father of, one more than ,plus, best friend, plus.(Takes one or more objects)

Ex : "Squares neighboring the Wumpus are smelly"

Objects : Wumpus, squares;

Function : smelly

Relation: neighboring

Syntax of FOL : Basic Elements

- **Constants** : Alice , Mumbai, 2, ...
- **Predicates**: brother, greater than, ...
- **Functions** : sqrt, leftlegof, ...
- **Variables** : x, y, a, b, ...
- **Connectives** : $\sim$, $\Leftrightarrow$, $\Rightarrow$, $\vee$, $\wedge$
- **Equality** : =
- **Quantifiers**: $\forall$, $\exists$

Symbols and Interpretations

- The basic syntactic elements of first-order logic are the symbols that stand for objects , relations and functions.
- Three kinds : **constant symbols** for objects, **predicate symbols** for relations and **function symbols** for functions.
- **Arity** : The number of arguments.

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{Predicate} \mid \mathit{Predicate}(\mathit{Term}, \ldots) \mid \mathit{Term} = \mathit{Term} \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
 & \mid & \neg\ \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \land \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \ \Rightarrow\ \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \ \Leftrightarrow\ \mathit{Sentence} \\
 & \mid & \mathit{Quantifier}\ \mathit{Variable}, \ldots\ \mathit{Sentence} \\
 & & \\
\mathit{Term} & \rightarrow & \mathit{Function}(\mathit{Term}, \ldots) \\
 & \mid & \mathit{Constant} \\
 & \mid & \mathit{Variable} \\
 & & \\
\mathit{Quantifier} & \rightarrow & \forall \mid \exists \\
\mathit{Constant} & \rightarrow & A \mid X_1 \mid \mathit{John} \mid \cdots \\
\mathit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\mathit{Predicate} & \rightarrow & \mathit{True} \mid \mathit{False} \mid \mathit{After} \mid \mathit{Loves} \mid \mathit{Raining} \mid \cdots \\
\mathit{Function} & \rightarrow & \mathit{Mother} \mid \mathit{LeftLeg} \mid \cdots \\
\text{OPERATOR PRECEDENCE} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

Unit – IV

Planning

Classical Planning

- **What is planning** : - The reasoning side of acting .
- Planning is the problem of finding a sequence of actions for achieving a goal from initial state where the actions are deterministic.
- For planning we need **domain description**, **action specification**, **goal description**.
- Each action has **preconditions & effects.**
- Effect can be positive or negative.

Planning is of two types

I. Classical planning

II. Non- classical planning

- **Classical planning** : When the environment is fully observable , deterministic, static , discrete.

- **Non-classical Planning** : When the environment is partially observable (i.e. the entire state of the environment is not visible at a given instant) or non-deterministic (i.e. the current state and chosen action cannot completely determine the next state of the environment). It involves different set of algorithms and agent designs.

General Purpose vs. Domain-Specific

- **General purpose**: symbolic descriptions of the problems and the domain. The plan generation algorithm the same.

  Advantage: - opportunity to have clear semantics.

  Disadvantage: - symbolic description requirement.

- **Domain Specific**: The plan generation algorithm depends on the particular domain.

  Advantage: - can be very efficient

  Disadvantage: - lack of clear semantics

State- and Plan-Space Planning

- **State-space** planners transform the state of the world. These planners search for a sequence of transformations linking the starting state and a final state.

- **Plan-space** planners transform the plans. These planners search for a plan satisfying certain conditions

Unit -V
Uncertainty

UNCERTAINTY : The agent can never be completely certain about the state of the external world since there is ambiguity and uncertainty.

Reasons for uncertainty:

- Sensors have limited precision, limited accuracy.
- Partial information
- Experimental errors.

Axioms of Probability:

1. $0 \leq P(a) \leq 1$: probabilities are between 0 and 1 inclusive
2. $P(true) = 1$, $P(false) = 0$
   probability of 1 for propositions believed to be absolutely true

   probability of 0 for propositions believed to be absolutely false
3. $P(a \cup b) = P(a) + P(b) - P(a \cup b)$

Bayes' theorem:

- Bayes' theorem is also known as Bayes' rule, Bayes' law, or Bayesian reasoning, which determines the probability of an event with uncertain knowledge.
- In probability theory, it relates the conditional probability and marginal probabilities of two random events. The Bayesian inference is an application of Bayes' theorem, which is fundamental to Bayesian statistics It is a way to calculate the value of $P(B|A)$ with the knowledge of $P(A|B)$.
- Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world

$$P(A|B) = P(B|A)P(A) / P(B)$$

where,

P(A) *and* ***P(B)*** *are the probabilities of events A and B*
P(A|B) *is the probability of event A when event B happens*
P(B|A) *is the probability of event B when A happens*

Example 1: A person has undertaken a job. The probabilities of completion of the job on time with and without rain are 0.44 and 0.95 respectively. If the probability that it will rain is 0.45, then determine the probability that the job will be completed on time.

Solution:

Let E1 be the event that the mining job will be completed on time and E2 be the event that it rains. We have

$P(A) = 0.45$,

$P(\text{no rain}) = P(B) = 1 - P(A) = 1 - 0.45 = 0.55$

By multiplication law of probability,

$P(E1) = 0.44$

$P(E2) = 0.95$

$P(E) = P(A)\,P(E1) + P(B)\,P(E2) = 0.45 \times 0.44 + 0.55 \times 0.95$

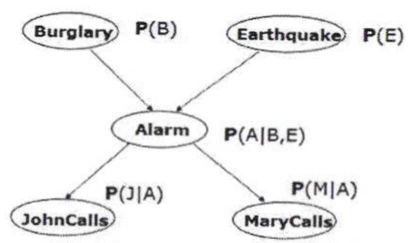
$= 0.198 + 0.5225 = 0.7205$

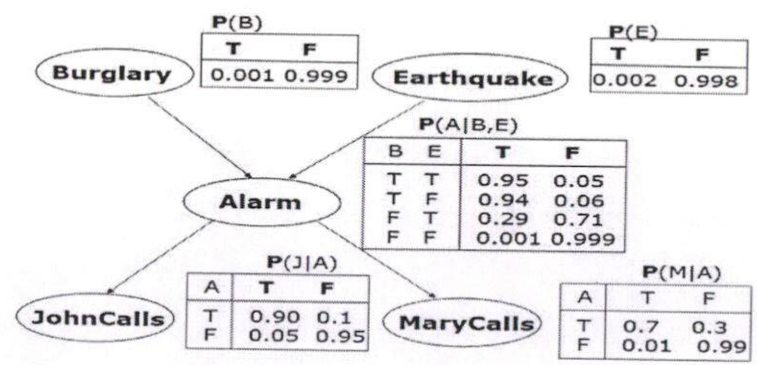
So, the probability that the job will be completed on time is 0.684.

Representing Knowledge in an Uncertain Domain — Bayesian Networks

Bayesian network is a data structure to represent an uncertain knowledge or domain. It is a directed acyclic graph whose nodes correspond to random variables.

Bayesian networks, often abbreviated as **“Bayes net”** were called **Belief networks.**





P(B)

T	F
0.001	0.999

P(E)

T	F
0.002	0.998

P(A|B,E)

B	E	T	F
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

P(J|A)

A	T	F
T	0.90	0.1
F	0.05	0.95

P(M|A)

A	T	F
T	0.7	0.3
F	0.01	0.99

Hall Ticket No. | | | | | | | | | | | Question Paper Code: A30530

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC AUTONOMOUS)

B.Tech __ Semester Regular End Examinations February -2021

(Regulation: CMRCET-R18)

Subject Name: ARTIFICIAL INTELLIGENCE

(Computer Science & Engineering)

Date: **Time: 3 hours** **Max.Marks:70**

PART-A

Answer all TEN questions (Compulsory)

Each question carries TWO marks. **10x2=20M**

1.	Define AI system and mention any two applications of AI	2 M	Remembering
2.	Outline percepts, actions, goals and environment for a taxi driver agent.	2 M	Understanding
3.	List the components of a game as a search problem	2 M	Remembering
4.	Show that the following sentences are inconsistent using propositional logic 1) If Jack misses many classes through illness, then he fails high school 2) If Jack fails high school, then he is uneducated	2 M	Evaluating
5.	List the quantifiers in first order logic	2 M	Remembering
6.	Differentiate propositional logic and first-order inference	2 M	Analyzing
7.	Explain briefly blocks world problem	2 M	Understanding
8.	Differentiate between search and planning	2 M	Analyzing
9.	Define Bayes rule	2 M	Remembering
10.	Define learning and list the important factors in learning	2 M	Remembering

PART-B

Answer any FIVE questions. One question from each unit either A or B (Compulsory)

Each question carries TEN Marks. **5x10=50M**

11. A.	Discuss in detail about utility based agents structure.	10M	Creating
	OR		
11. B.	1) Compare and contrast uninformed search strategies 2) Explain A* heuristic search algorithm	5M 5M	Analyzing Understanding
12. A.	Demonstrate the function of alpha-beta pruning with its effectiveness	10M	Understand
	OR		
12. B.	1) Differentiate between proposition logic& predicate logic with examples 2) Discuss in brief about forward chaining in propositional logic	5M 5M	Analyzing Creating
13. A.	Model the structure of a completeness proof for resolution in first order logic	10M	Apply

OR			
13. B.	1) Discuss briefly properties of good and bad knowledge bases 2) Differentiate knowledge engineering Vs. programming	5M 5M	Create Analyze
14. A.	Discuss in brief about most popular and effective classical planning approaches	10M	Creating
OR			
14. B.	Compare conformant planning and contingency planning with examples	10M	Analyzing
15. A.	Create an algorithm for construction of Bayesian net	10M	Appling
OR			
15. B.	Explain Supervised learning with example.	10M	Understanding

(You are requested to give division of marks in the marks column, If any sub questions are given in any question.)

NOTE:

1. Please see that the questions are within the scope of the syllabus, which can be written by Students in allotted 3 hours time.

Hall Ticket No. | | | | | | | | | | | Question Paper Code: A30530

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC AUTONOMOUS)

B.Tech V Semester Regular Examinations December-2022
(Regulation: CMRCET-R18)

Subject Name: ARTIFICIAL INTELLIGENCE
(CSE)

Date: **Time: 3 hours** **Max.Marks:70**

PART-A
Answer all TEN questions (Compulsory)
Each question carries TWO marks. **10x2=20M**

1.	Give one definition of AI.	2 M	Remember
2.	Define intelligent agent.	2 M	Remember
3.	What are the minimum number of colors required to color any map in this world?	2 M	Remember
4.	What is binary constraint?	2 M	Remember
5.	Define universal quantifiers.	2 M	Remember
6.	Define semantic networks.	2 M	Remember
7.	What is blocks world problem?	2 M	Remember
8.	Define critical path in job scheduling problems.	2 M	Remember
9.	What is unsupervised learning method.	2 M	Remember
10.	What is a monte carlo algorithm?	2 M	Remember

PART-B
Answer any FIVE questions. One question from each unit either A or B (Compulsory)
Each question carries TEN Marks. **5x10=50M**

11. A.	Give four definitions of Artificial intelligence system and Justify the statement Agent=Architecture + Program	10M	Understand
	OR		
11. B.	Explain depth limited search with time and space complexity and how heuristic functions are useful to solve 8 Puzzle problem.	10M	Understand
12. A.	Solve the constraint satisfaction problem. T W O T W O (+) -------------------- F O U R --------------------	10M	Apply

	OR		
12. B.	**Solve the Wumpus world problem with neat diagrams using Logic programming(prepositional logic or predicate logic).**	10M	Apply
13. A.	**Represent the following statements and Solve the problem using resolution proof.** **S1: Everyone who loves all animals is loved by someone.** **S2: Anyone who kills an animal is loved by no one.** **S3: Jack loves all animals.** **S4: Either Jack or Curiosity killed the cat, who is named Tuna.** **S5: Did Curiosity kill the cat?**	10M	Apply
	OR		
13. B.	**Explain semantic networks and description logics with example.**	10M	Understand
14. A.	**Discuss How to solve the planning problems with state space search.**	10M	Understand
	OR		
14. B.	**Discuss how to solve hierarchical task network problems.**	10M	Understand
15. A.	**How to represent conditional distributions efficiently with example.**	10M	Apply
	OR		
15. B.	**Apply Inference using full joint distribution for the Figure 13.3 and find** **(i) P(cavity or toothache)** **(ii) Marginal probability of Cavity** **(iii) P(cavity|toothache)** **(iv) P(no cavity|toothache)**	10M	Apply

	toothache		*¬toothache*	
	catch	*¬catch*	*catch*	*¬catch*
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

Figure 13.3 A full joint distribution for the *Toothache*, *Cavity*, *Catch* world.

Hall Ticket No. | | | | | | | | | | | **Question Paper Code: A30530**

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC AUTONOMOUS)

B.Tech V Semester Regular Examinations December-2022

(Regulation: CMRCET-R18)

Subject Name: ARTIFICIAL INTELLIGENCE

(CSE)

Date: **Time: 3 hours** **Max.Marks:70**

PART-A

Answer all TEN questions (Compulsory)

Each question carries TWO marks. **10x2=20M**

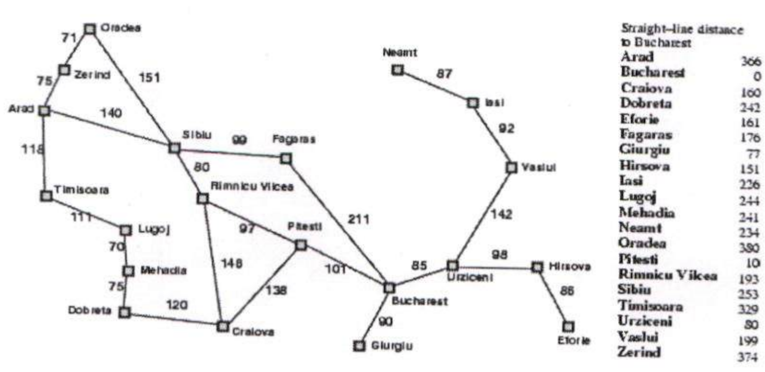
1.	Define artificial intelligence system.	2 M	Remember
2.	Define PEAS in agent system.	2 M	Remember
3.	Define pruning in a game tree.	2 M	Remember
4.	What is utility function?	2 M	Remember
5.	Define propositional logic.	2 M	Remember
6.	What is first order logic.	2 M	Remember
7.	Define classical planning problems.	2 M	Remember
8.	What is a closed world assumptions.	2 M	Remember
9.	Define a random variables.	2 M	Remember
10.	What is a decision tree method?	2 M	Remember

PART-B

Answer any FIVE questions. One question from each unit either A or B (Compulsory)

Each question carries TEN Marks. **5x10=50M**

11. A.	Apply depth first search and Greedy best first search to find the route from Arad to Bucharest in the figure shown below.	10M	Apply



OR

11. B.	Apply Breadth first search and A* search to find the route from Arad to Bucharest in the above figure.	10M	Apply
12. A.	Explain Arc Consistency3 algorithm of graph coloring problem with example .	10M	Understand

	OR		
12. B.	**Explain Min Max Algorithm with example for 2 player games.**	10M	Understand
13. A.	**Explain the models for first order logic and its syntax using Backus naur form.**	10M	Understand
	OR		
13. B.	**Explain the unification algorithm with example.**	10M	Understand
14. A.	**Explain the language of planning problems.**	10M	Remember
	OR		
14. B.	**Explain the job shop scheduling problem for assembling two cars.**	10M	Remember
15. A.	**Discuss the semantics of bayesian networks with example.**	10M	Understand
	OR		
15. B.	**Explain random variables, prior probability and conditional probability with example.**	10M	Understand

Hall Ticket No. | | | | | | | | | | | Question Paper Code: A30530

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC AUTONOMOUS)

B.Tech __ Semester Regular End Examinations February -2021
(Regulation: CMRCET-R18)

Subject Name: ARTIFICIAL INTELLIGENCE

(Computer Science & Engineering)

Date: **Time: 3 hours** **Max.Marks:70**

PART-A

Answer all TEN questions (Compulsory)
Each question carries TWO marks. **10x2=20M**

1.	Demonstrate with a neat sketch how the agents interact with environments through sensors and effectors.	2 M	Understanding
2.	List the basic elements of an agent for well define problems.	2 M	Remembering
3.	Illustrate in brief a generic knowledge-based agent.	2 M	Understanding
4.	Interpret the following sentences in the predicate logic 1) Paris and Marseilles are both in France. 2) There is a country that borders both Iraq and Pakistan.	2 M	Evaluating
5.	Explain briefly kinship domain.	2 M	Understanding
6.	Represent the following sentences in first-order logic, using a consistent vocabulary (which you must define): 1) Not all students take both History and Biology. 2) Only one student failed History.	2 M	Evaluating
7.	Define classical planning with example.	2 M	Remembering
8.	Differentiate bounded indeterminacy and unbounded indeterminacy.	2 M	Analyzing
9.	Compare core AI Vs probabilistic AI.	2 M	Analyzing
10.	Define pruning and list issues in decision trees.	2 M	Remembering

PART-B

Answer any FIVE questions. One question from each unit either A or B (Compulsory)
Each question carries TEN Marks. **5x10=50M**

11. A.	Discuss in detail about goal based agents structure.	10M	Creating
	OR		
11. B.	Demonstrate Depth-first iterative deepening algorithm with an example and list advantages and disadvantages of it over depth-first search algorithm.	10M	Understanding
12. A.	Model a typical Wumpus world environment with its actions and reasoning.	10M	Applying
	OR		
12. B.	Discuss in detail about backward chaining and how it differentiates from forward chaining in propositional logic.	10M	Creating & Analyzing
13. A.	Discuss in detail syntax and semantics of first order logic.	10M	Creating

	OR		
13. B.	1) Discuss the importance of top level ontology of the world 2) Explain composite objects with examples.	6M 4M	Creating Understanding
14. A.	Differentiate forward state space search and backward (regression) relevant-states search.	10M	Analyzing
	OR		
14. B.	Explain multi agent planning with example.	10M	Understanding
15. A.	Justify how uncertainty and ignorance can be handled using Dempster-Shafer theory.	10M	Evaluating
	OR		
15. B.	Demonstrate inductive logic programming in detail.	10M	Understanding

(You are requested to give division of marks in the marks column, If any sub questions are given in any question.)

NOTE:

1. Please see that the questions are within the scope of the syllabus, which can be written by Students in allotted 3 hours time.