ADUSUMILLI GOPALAKRISHNAIAH & SUGARCANE GROWERS

SIDDHARTHA DEGREE COLLEGE OF ARTS & SCIENCE

Vuyyuru-521 165, Krishna District, Andhra Pradesh

An Autonomous College in the Jurisdiction of Krishna University

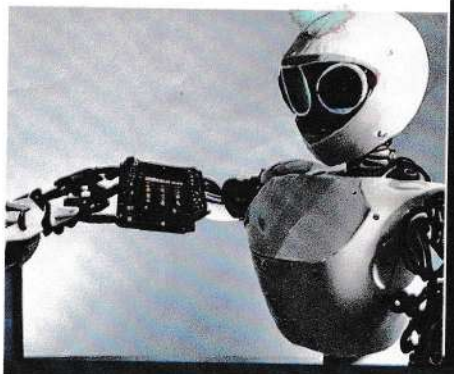
Accredited by NAAC with "A" Grade

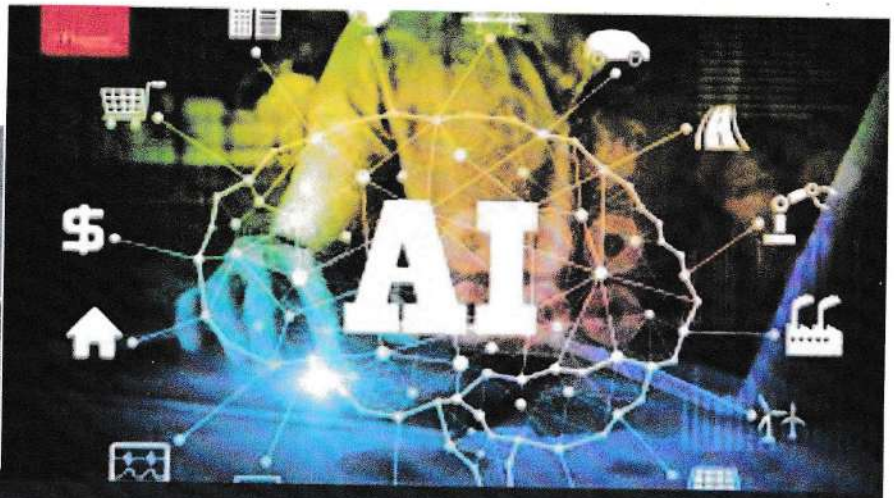
VAD COURSE :ARTIFICIAL INTELIGENCE

VAC CODE:AIVAC101

CLASS :IIBSc(MPCs)

DURATION :30 DAYS





DEPARTEMENT OF COMPUTER SCIENCE

Contact

www.agsgsc.edu.in

agsgsiddhartha@gmail.com

08676-233267

Address:

Door no 2.391
College Road,Near Kota Complex,
Vuyyuru-521165

A.G. & S.G. Siddhartha Degree College of Arts & Science

Vuyyuru-521165, Krishna District, Andhra Pradesh

(Managed by: Siddhartha Academy of General & Technical Education, Vijayawada-10)

An Autonomous College in the Jurisdiction of Krishna University

Accredited by NAAC with "A" Grade

DEPARTMENT OF COMPUTER SCIENCE

Value Added Course

Title: ARTIFICIAL INTELLIGENCE

Name of the Lecturer	:	Gona Katyayini
Class	:	II MPC's
Duration of the Course	:	30 HOURS
VAC Code	:	AIVAC101

A.G. & S.G. Siddhartha Degree College of Arts & Science

Vuyyuru-521165, Krishna District, Andhra Pradesh

Value Added Course

Title: ARTIFICIAL INTELLIGENCE

Objectives : 1)Achieve the objectives of company XX by 20202

2)Boost organizational Performance at all levels

3)Use an integrated smart digital system that can overcome challenges and provide quick efficient solution.

4)Make company XX the first in the field of Al investments various sectors

Methodology :Teacher-Centered method

Duration : 30 Hours

A.G. & S.G. Siddhartha Degree College of Arts & Science

Vuyyuru-521165, Krishna District, Andhra Pradesh

Value Added Course

Title: ARTIFICIAL INTELLIGENCE

Date From 18/11/2021 to 12/01/2022

Date	Content	Module No.
18/11/21	**INTRODUCTION TO AI AND PRODUCTION SYSTEMS** Introduction to Al-Problem formulation, Control strategies, Search strategies, Problem characteristics,	I
01/12/21	**REPRESENTATION OF KNOWLEDGE** Game playing Knowledge representation, Knowledge representation using Predicate logic, Introduction to predicate calculus, Use of predicate calculus	II
10/12/21	**KNOWLEDGE INFERENCE** Knowledge representation -Production based system, Frame based system. Inference - Backward chaining, Forward chaining	III
04/01/22	**PLANNING AND MACHINE LEARNING** Basic plan generation systems , Advanced plan generation systems ,K strips ,Strategic explanations	IV

A.G. & S.G. Siddhartha Degree College of Arts & Science

Vuyyuru-521165, Krishna District, Andhra Pradesh

Value Added Course
Student Enrolment Sheet

Class :II MPC's

S. No	Roll No.	Name of the Student	Signature
1	20.601	Chillimuntha Lokesh	C. Lokesh
2	20.602	Loya Naga Padma	L.N. Padma
3	20.604	Pulipaka Chakradhar	P. Chakradhar
4	20.605	Alaparthi Kumar Venkata Sai Kiran	A.K.V. Saikiran
5	20.606	Elusuri Poojitha	E. Poojitha
6	20.607	Rebba Bhanu Sree	R. Bhanu Sree
7	20.608	Busi Manjusha	B. Manjusha
8	20.609	Gopalajoshula Prathyusha	G. Prathyusha
9	20.610	Kolluri Likitha	K. Likitha
10	20.611	Kurapati Harika	K. Harika
11	20.612	Goriparthi Sampoorna	G. Sampoorna
12	20.613	Kolli Naga Tirumala Reddy	K. Naga Tirumala Reddy
13	20.614	Namala V V Sandhya	N V V Sandhya
14	20.615	Tumuluri Yogambika	T. Yogambika
15	20.616	Arilli Veeramma	A. Veeramma
16	20.617	Gangireddy Jahnavi	G. Jahnavi

17	20.618	Katta Pravallika	K. Pravallika
18	20.619	Kagitha Hinduja	K. Hinduja
19	20.620	Ede Sravani Durga Devi	E.S.D. Devi
20	20.621	Munipalli Ajay	M. Ajay
21	20.622	Dasari Naga Mallika	D. Naga mallika
22	20.624	Juvvanapudi Sai Kiran	J. Sai Kiran
23	20.625	Taviti Ratna Kumari	T. Ratnakumari
24	20.626	Parimi Prem Chand	P. Prem chand
25	20.628	Shaik Sabiha Tasleem	S. Sabiha Taleem
26	20.629	Vasadi Radha	V. Radha.
27	20.630	Chittibomma Poorna Chandra Moulika	Ch. P. C. Moulika
28	20.631	Dokku Gnana Prasanna	D. Gnana Prasanna
29	20.633	Dabbada Sai	D. Sai
30	20.634	Mareedu Aliveni	M. Aliveni

Signature of Lecturer

Signature of HOD

PRINCIPAL
AG & SG Siddhartha Degree College of
Arts&Science (Autonomous),Vuyyuru

ARTIFICIAL INTELLIGENCE
UNIT-1:

Introduction to AI and production system

- **Introduction to AI - problem formulation**

Problem formation is the step in problem definition that is used to understand and decide a course of action that needs to be considered to achieve a goal. If there is more than one way an agent can reach its goal, then it causes complexity in terms of actually achieving the goal as there would be too many steps and paths that the AI entity can take to reach the goal that it causes confusion and a massive drop in the efficiency at the same time. Problem formulation can be done in many steps such as the definition of the initial state of the agent, determining possible actions that the agent can take, creation of transition models to describe the actions of the agent. Testing the goal and determining the path cost function are also important complements in determining the problem formulation approach.

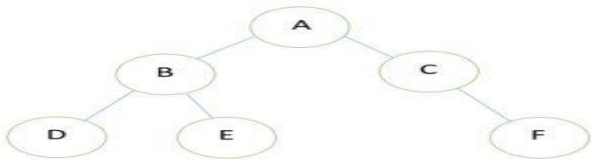
Cases involving Artificial Intelligence Issues

Artificial intelligence is being used by programmers all around the world to automate systems for effective both resource and time management. Games and puzzles can pose some of the most frequent issues in daily life. The use of ai algorithms may effectively tackle this. Various problem-solving methods are implemented to create solutions for a variety complex puzzles, includes mathematics challenges such crypto-arithmetic and magic squares, logical puzzles including Boolean formulae as well as N-Queens, and quite well games like Sudoku and Chess. Therefore, these below represent some of the most common issues that artificial intelligence has remedied:

- Chess
- N-Queen problem
- Tower of Hanoi Problem
- Travelling Salesman Problem
- Water-Jug Problem

- **Control strategies**

Control Strategy in Artificial Intelligence scenario is a technique or strategy, tells us about which rule has to be applied next while searching for the solution of a problem within problem space. It helps us to decide which rule has to apply next without getting stuck at any point. These rules decide the way we approach the problem and how quickly it is solved and even whether a problem is finally solved.
Control Strategy helps to find the solution when there is more than one rule or fewer rules for finding the solution at each point in problem space. A good Control strategy has two main characteristics:
Examples:



Breadth-First Search: It searches along the breadth and follows first-in-first-out queue data structure approach. It will start to scan node A first and then B-C-D-E-F.
Depth-First Search: It searches along the depth and follows the stack approach. The sequence for scanning nodes will be A-B-D-E-C-F, it scans all the sub-nodes of parent nodes and then moves to another node.
Widely used Control Strategies are Breadth-First Search, Depth-First Search, Generate and Test, Hill-Climbing, Best-first search, Problem Reduction and many more.

- **Search strategies**

What are search strategies in AI?

- Breadth-First Search. ...
- Depth-First Search. ...
- Bidirectional Search. ...
- Uniform Cost Search. ...
- Iterative Deepening Depth-First Search. ...
- Comparison of Various Algorithms Complexities. ...
- Heuristic Evaluation Functions. ...
- Pure Heuristic Search.

Informed Search Strategy

- **Informed search strategy further includes two searching techniques. These are:**
  1. **A* Search Technique.**
  2. **AO* Search Technique.**

A* Search Technique

- **A* search technique is an informal search strategy but can be called as a form of best first search.**
- **It is a search technique which the most optimistic node is expanded by expanding a graph.**
- **The node of the graph can be evaluated by using two functions i.e. g(n) and h(n).**

Here,
g(n) = Cost/Distance to reach node "n".
h(n) = Cost/Distance to reach from node "n" to the goal node.

- **For evaluating any node, function f(n) is generated and used as:**

f(n) = g(n) + h(n).
where,
f(n) = Estimated cost/distance of solution through node "n".

Un-Informed Search Strategy

- **Un-Informed search strategy further includes two techniques. These are:**
  1. **Breadth First Search.**
  2. **Depth First Search.**

➢ Problem characteristics

Problem characteristics in AI refers to finding an optimal way and a good solution to characterize the problem. In addition, problem solving is one of the key concerns in AI. Certainly, we know AI is a vast field and branch of computer science, there can be complex problems to solve.

Let us take a look at some of the major problem characteristics.

MAJOR PROBLEM CHARACTERISTICS

Firstly, we need to know if the problem is decomposable or not. For instance, block world problem or Tower of Hanoi are easily decomposable. In Tower of Hanoi some rules are define to move the disks from source to destination. Moreover, each movement is divided into steps.

Secondly, problem is categorize into following:

- Ignorable
- Recoverable
- Irrecoverable

Ignorable problems are those which can be solved using simple control structure. For instance, mathematical problems.Recoverable problems are those where we can use backtracking to solve the problem. For instance 8-puzzle problem.

PROBLEM CHARACTERISTICS STEPS

- Define the problem precisely.
- Give the initial input required.
- Apply knowledge
- Choose best optimal technique for problem solving.

UNIT-2:

REPRESENTATION OF KNOWLEDGE

➢ Game playing knowledge representation

Game Playing is an important domain of artificial intelligence. Games don't require much knowledge; the only knowledge we need to provide is the rules, legal moves and the conditions of winning or losing the game. Both players try to win the game. So, both of them try to make the best move possible at each turn. Searching techniques like BFS(Breadth First Search) are not accurate for this as the branching factor is very high, so searching will take a lot of time. So, we need another search procedures that improve –

- **Generate procedure** so that only good moves are generated.
- **Test procedure** so that the best move can be explored first.

Game playing is a popular application of artificial intelligence that involves the development of computer programs to play games, such as chess, checkers, or Go. The goal of game playing in artificial intelligence is to develop algorithms that can learn how to play games and make decisions that will lead to winning outcomes.

1. One of the earliest examples of successful game playing AI is the chess program Deep Blue, developed by IBM, which defeated the world champion Garry Kasparov in 1997. Since then, AI has been applied to a wide range of games, including two-player games, multiplayer games, and video games.

There are two main approaches to game playing in AI, rule-based systems and machine learning-based systems.

1. **Rule-based systems** use a set of fixed rules to play the game.
2. **Machine learning-based systems** use algorithms to learn from experience and make decisions based on that experience.

In recent years, machine learning-based systems have become increasingly popular, as they are able to learn from experience and improve over time, making them well-suited for complex games such as Go. For example, AlphaGo, developed by DeepMind, was the first machine learning-based system to defeat a world champion in the game of Go.

Advantages of Game Playing in Artificial Intelligence:

1. **Advancement of AI:** Game playing has been a driving force behind the development of artificial intelligence and has led to the creation of new algorithms and techniques that can be applied to other areas of AI.
2. **Education and training:** Game playing can be used to teach AI techniques and algorithms to students and professionals, as well as to provide training for military and emergency response personnel.
3. **Research:** Game playing is an active area of research in AI and provides an opportunity to study and develop new techniques for decision-making and problem-solving.
4. **Real-world applications:** The techniques and algorithms developed for game playing can be applied to real-world applications, such as robotics, autonomous systems, and decision support systems.

➢ Konwledge representation using predicated logic

Introduction

Predicate Logic in AI is fundamentally a method for describing and modifying assertions about objects and their characteristics. It includes a collection of rules and symbols that enable us to build complex statements from simpler ones.

Predicates and variables are the core components of Predicate Logic in AI. A predicate is an assertion made regarding one or more things. For instance, the predicate "is blue" declares that a specific item possesses the attribute of being blue. A variable serves as a stand-in for an object so that assertions can apply to any object of a particular type. We can generalize assertions about vehicles by using the variable ﷺ to represent any car, for instance. There must be at least one object (represented by the variable ﷺ) with the quality of being blue, according to the adage "exists ﷺ, ﷺ is blue".

Characteristics of Predicate Logic

Predicate Logic in AI has several characteristics that make it a powerful tool for AI applications. Some of these characteristics are:

- The Logical inference is allowed.
- More accurate knowledge representation of facts of the real world.
- Program designing is its application area.
- Better theoretical foundation.
- A predicate with no variable is called a Ground Atom.

➢ Introduction to predicated calculus

predicate calculus, also called **Logic Of Quantifiers**, that part of modern formal or symbolic logic which systematically exhibits the logical relations between sentences that hold purely in virtue of the manner in which predicates or noun expressions are distributed through ranges of subjects by means of quantifiers such as “all” and “some” without regard to the meanings or conceptual contents of any predicates in particular. Such predicates can include both qualities and relations; and, in a higher-order form called the functional calculus, it also includes functions, which are “framework” expressions with one or with several variables that acquire definite truth-values only when the variables are replaced by specific terms. The predicate calculus is to be distinguished from the propositional calculus, which deals with unanalyzed whole propositions related by connectives (such as “and,” “if . . . then,” and “or”).

The traditional syllogism is the most well-known sample of predicate logic, though it does not exhaust the subject. In such arguments as “All C are B and no B are A, so no C are A,” the truth of the two premises requires the truth of the conclusion in virtue of the manner in which the predicates B and A are distributed with reference to the classes specified by C and B, respectively. If, for example, the predicate A belonged to only one of the B’s, the conclusion then could possibly be false—some C could be an A.

More From Britannica

formal logic: The predicate calculus

Modern symbolic logic, of which the predicate calculus is a part, does not restrict itself, however, to the traditional syllogistic forms or to their symbolisms, a very large number of which have been devised. The predicate calculus usually builds upon some form of the propositional calculus. It then proceeds to give a classification of the sentence types that it contains or deals with, by reference to the different manners in which predicates may be distributed within sentences. It distinguishes, for example, the following two types of sentences: "All *F*'s are either *G*'s or *H*'s," and "Some *F*'s are both *G*'s and *H*'s." The conditions of truth and falsity in the basic sentence types are determined, and then a cross-classification is made that groups the sentences formulable within the calculus into three mutually exclusive classes—(1) those sentences that are true on every possible specification of the meaning of their predicate signs, as with "Everything is *F* or is not *F*"; (2) those false on every such specification, as with "Something is *F* and not *F*"; and (3) those true on some specifications and false on others, as with "Something is *F* and is *G*."

- **Use of predicate calculus**

Predicate

A predicate is an expression of one or more variables defined on some specific domain. A predicate with variables can be made a proposition by either assigning a value to the variable or by quantifying the variable.

Consider the following statement.

Ram is a student.

Now consider the above statement in terms of Predicate calculus.

Here "is a student" is a predicate and Ram is subject.
Let's denote "Ram" as x and "is a student" as a predicate P then we can write the above statement as P(x).
Generally a statement expressed by Predicate must have at least one object associated with Predicate. In our case, Ram is the required object with associated with predicate P.

Well Formed Formula

Well Formed Formula (wff) is a predicate holding any of the following −

All propositional constants and propositional variables are wffs

If x is a variable and Y is a wff, ∀x Y and ∀x Y are also wff

Truth value and false values are wffs

Each atomic formula is a wff

All connectives connecting wffs are wffs

Free and Bound variables

Consider a Predicate formula having a part in form of $(\exists x)\ P(x)$ of $(x)P(x)$, then such part is called x-bound part of the formula. Any occurrence of x in x-bound part is termed as bound occurrence and any occurrence of x which is not x-bound is termed as free occurrence. See the examples below -

$(\exists x)\ (P(x) \wedge Q(x))$

$(\exists x)\ P(x) \wedge Q(x)$

Universe of Discourse

We can limit the class of individuals/objects used in a statment. Here limiting means confining the input variable to a set of particular individuals/objects. Such a restricted class is termed as Universe of Discourse/domain of individual or universe. See the example below:

Some cats are black.

$C(x)$: x is a cat.

$B(x)$: x is black.

$(\exists x)(C(x) \wedge B(x))$

UNIT-3

KNOWLEDGE INFERENCE

Knowledge representation -

- **Production based on system**

What is Production System?

Production system or production rule system is a computer program typically used to provide some form of artificial intelligence, which consists primarily of a set of rules about behavior but it also includes the mechanism necessary to follow those rules as the system responds to states of the world.

Components of Production System

The major components of the Production System in Artificial Intelligence are:

- **Global Database:** The global database is the central data structure used by the production system in Artificial Intelligence.
- **Set of Production Rules:** The production rules operate on the global database. Each rule usually has a precondition that is either satisfied or not by the global database. If the precondition is satisfied, the rule is usually be applied. The application of the rule changes the database.
- **A Control System:** The control system then chooses which applicable rule should be applied and ceases computation when a termination condition on the database is satisfied. If multiple rules are to fire at the same time, the control system resolves the conflicts.

Features of Production System in Artificial Intelligence

The main features of the production system include:

1. Simplicity: The structure of each sentence in a production system is unique and uniform as they use the "IF-THEN" structure. This structure provides simplicity in knowledge representation. This feature of the production system improves the readability of production rules.

2. Modularity: This means the production rule code the knowledge available in discrete pieces. Information can be treated as a collection of independent facts which may be added or deleted from the system with essentially no deleterious side effects.

3. Modifiability: This means the facility for modifying rules. It allows the development of production rules in a skeletal form first and then it is accurate to suit a specific application.

4. Knowledge-intensive: The knowledge base of the production system stores pure knowledge. This part does not contain any type of control or programming information. Each production rule is normally written as an English sentence; the problem of semantics is solved by the very structure of the representation.

- **Frame based system**

Frames provide a structure for representing knowledge that can be used by AI systems to reason about the world. They can also be used to store and retrieve information from memory, and to make inferences about new situations. Frames can also be used to represent plans and goals, and to generate new actions.

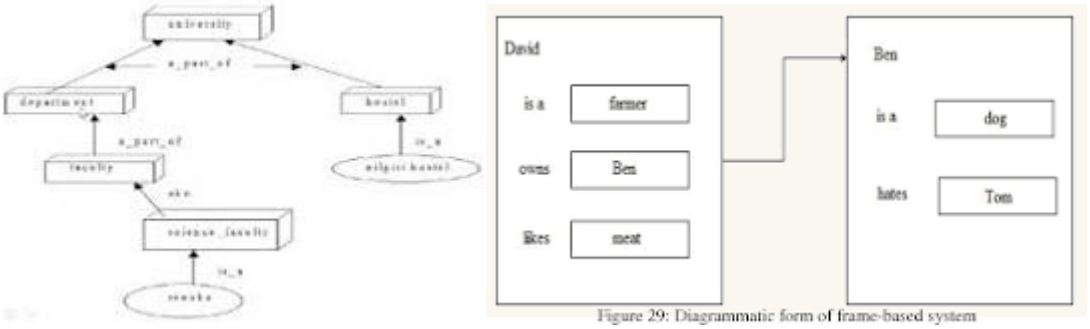


Figure 29: Diagrammatic form of frame-based system

The frame contains information on how to use the frame, what to expect next, and what to do when these expectations are not met. Some information in the frame is generally unchanged while other information, stored in "terminals", usually change. Terminals can be considered as variables. Top-level frames carry information, that is always true about the problem in hand, however, terminals do not have to be true. Their value might change with the new information encountered. Different frames may share the same terminals.

Each piece of information about a particular frame is held in a slot. The information can contain:

Facts or DataValues (called facets)

Procedures (also called procedural attachments)

- IF-NEEDED: deferred evaluation
- IF-ADDED: updates linked information

Default Values

- For Data
- For Procedures

Features and advantages

A frame's terminals are already filled with default values, which is based on how the human mind works. For example, when a person is told "a boy kicks a ball", most people will visualize a particular ball (such as a familiar soccer ball) rather than imagining some abstract ball with no attributes.

One particular strength of frame-based knowledge representations is that, unlike semantic networks, they allow for exceptions in particular instances. This gives frames an amount of flexibility that allows representations of real-world phenomena to be reflected more accurately.

Like semantic networks, frames can be queried using spreading activation. Following the rules of inheritance, any value given to a slot that is inherited by subframes will be updated (IF-ADDED) to the corresponding slots in the subframes and any new instances of a particular frame will feature that new value as the default.

Because frames are based on structures, it is possible to generate a semantic network given a set of frames even though it lacks explicit arcs. References to Noam Chomsky and his generative grammar of 1950 are generally missing from Minsky's work.

The simplified structures of frames allow for easy analogical reasoning, a much prized feature in any intelligent agent. The procedural attachments provided by frames also allow a degree of flexibility that makes for a more realistic representation and gives a natural affordance for programming applications.

- **Backward chaining**

In artificial intelligence, forward and backward chaining is one of the important topics, but before understanding forward and backward chaining lets first understand that from where these two terms came.

Inference engine:

The inference engine is the component of the intelligent system in artificial intelligence, which applies logical rules to the knowledge base to infer new information from known facts. The first inference engine was part of the expert system. Inference engine commonly proceeds in two modes, which are:

a. **Forward chaining**
b. **Backward chaining**

Horn Clause and Definite clause:

Horn clause and definite clause are the forms of sentences, which enables knowledge base to use a more restricted and efficient inference algorithm. Logical inference algorithms use forward and backward chaining approaches, which require KB in the form of the **first-order definite clause**

Definite clause: A clause which is a disjunction of literals with **exactly one positive literal** is known as a definite clause or strict horn clause.

Horn clause: A clause which is a disjunction of literals with **at most one positive literal** is known as horn clause. Hence all the definite clauses are horn clauses.

Example: (¬ p V ¬ q V k). It has only one positive literal k.

It is equivalent to p ∧ q → k.

Backward Chaining:

Backward-chaining is also known as a backward deduction or backward reasoning method when using an inference engine. A backward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through rules to find known facts that support the goal.

Properties of backward chaining:

- It is known as a top-down approach.
- Backward-chaining is based on modus ponens inference rule.
- In backward chaining, the goal is broken into sub-goal or sub-goals to prove the facts true.
- It is called a goal-driven approach, as a list of goals decides which rules are selected and used.
- Backward -chaining algorithm is used in game theory, automated theorem proving tools, inference engines, proof assistants, and various AI applications.
- The backward-chaining method mostly used a **depth-first search** strategy for proof.

Example: In backward-chaining, we will use the same above example, and will rewrite all the rules

- American (p) ∧ weapon(q) ∧ sells (p, q, r) ∧ hostile(r) → Criminal(p) ...(1)
  Owns(A, T1) .....(2)
- Missile(T1)
- ?p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A) .....(4)
- Missile(p) → Weapons (p) .....(5)
- Enemy(p, America) →Hostile(p) .....(6)
- Enemy (A, America) .....(7)
- American(Robert). .....(8)

- **Forward chaining**

Inference engine:

The inference engine is the component of the intelligent system in artificial intelligence, which applies logical rules to the knowledge base to infer new information from known facts. The first inference engine was part of the expert system. Inference engine commonly proceeds in two modes, which are:

a. **Forward chaining**
b. **Backward chaining**

A. Forward Chaining

Forward chaining is also known as a forward deduction or forward reasoning method when using an inference engine. Forward chaining is a form of reasoning which start with atomic sentences in the knowledge base and applies inference rules (Modus Ponens) in the forward direction to extract more data until a goal is reached.

The Forward-chaining algorithm starts from known facts, triggers all rules whose premises are satisfied, and add their conclusion to the known facts. This process repeats until the problem is solved.

Properties of Forward-Chaining:

- It is a down-up approach, as it moves from bottom to top.
- It is a process of making a conclusion based on known facts or data, by starting from the initial state and reaches the goal state.
- Forward-chaining approach is also called as data-driven as we reach to the goal using available data.
- Forward -chaining approach is commonly used in the expert system, such as CLIPS, business, and production rule systems.

Example:

"As per the law, it is a crime for an American to sell weapons to hostile nations. Country A, an enemy of America, has some missiles, and all the missiles were sold to it by Robert, who is an American citizen."

Prove that **"Robert is criminal."**

To solve the above problem, first, we will convert all the above facts into first-order definite clauses, and then we will use a forward-chaining algorithm to reach the goal.

UNIT-4

PLANNING AND MACHINE LEARNING

➢ Basic plan generation system

Artificial intelligence is an important technology in the future. Whether it is intelligent robots, self-driving cars, or smart cities, they will all use different aspects of artificial intelligence!!! But Planning is very important to make any such AI project.

Even Planning is an important part of Artificial Intelligence which deals with the tasks and domains of a particular problem. Planning is considered the logical side of acting.

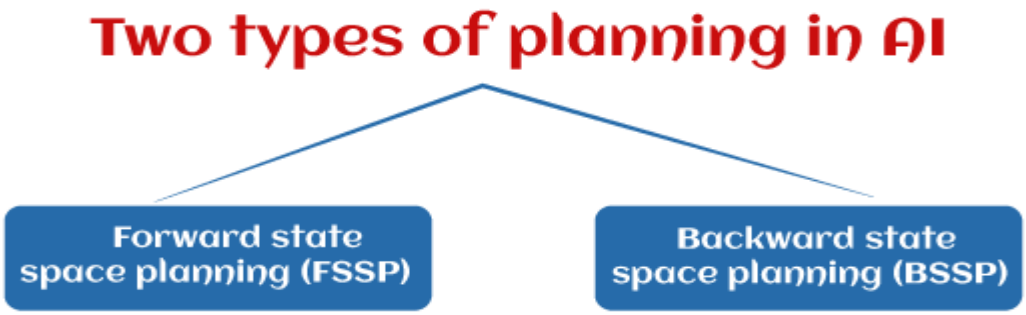
Everything we humans do is with a definite goal in mind, and all our actions are oriented towards achieving our goal. Similarly, Planning is also done for Artificial Intelligence.

For example, Planning is required to reach a particular destination. It is necessary to find the best route in Planning, but the tasks to be done at a particular time and why they are done are also very important.

What is a Plan?

We require domain description, task specification, and goal description for any planning system. A plan is considered a sequence of actions, and each action has its preconditions that must be satisfied before it can act and some effects that can be positive or negative.

So, we have **Forward State Space Planning (FSSP)** and **Backward State Space Planning (BSSP)** at the basic level.



1. Forward State Space Planning (FSSP)

FSSP behaves in the same way as forwarding state-space search. It says that given an initial state S in any domain, we perform some necessary actions and obtain a new state S' (which also contains some new terms), called a progression. It continues until we reach the target position. Action should be taken in this matter.

- **Disadvantage**: Large branching factor
- **Advantage**: The algorithm is Sound

2. Backward State Space Planning (BSSP)

BSSP behaves similarly to backward state-space search. In this, we move from the target state g to the sub-goal g, tracing the previous action to achieve that goal. This process is called regression (going back to the previous goal or sub-goal). These sub-goals should also be checked for consistency. The action should be relevant in this case.

- **Disadvantages**: not sound algorithm (sometimes inconsistency can be found)
- **Advantage**: Small branching factor (much smaller than FSSP)

So for an efficient planning system, we need to combine the features of FSSP and BSSP, which gives rise to target stack planning which will be discussed in the next article.

What is planning in AI?

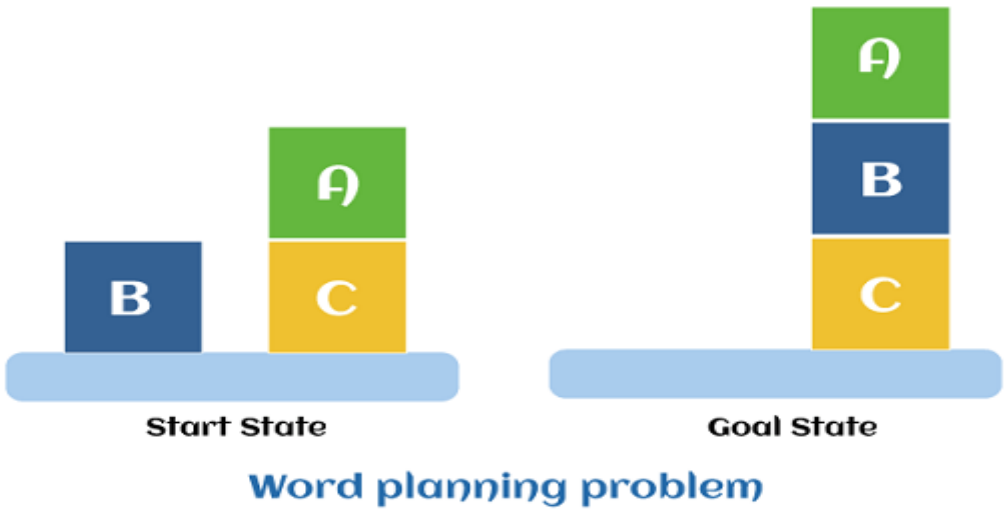
Planning in artificial intelligence is about decision-making actions performed by robots or computer programs to achieve a specific goal.

Execution of the plan is about choosing a sequence of tasks with a high probability of accomplishing a specific task.

Block-world planning problem

- The block-world problem is known as the Sussmann anomaly.
- The non-interlaced planners of the early 1970s were unable to solve this problem. Therefore it is considered odd.
- When two sub-goals, G1 and G2, are given, a non-interleaved planner either produces a plan for G1 that is combined with a plan for **G2** or vice versa.
- In the block-world problem, three blocks labeled 'A', 'B', and 'C' are allowed to rest on a flat surface. The given condition is that only one block can be moved at a time to achieve the target.

The start position and target position are shown in the following diagram.



Word planning problem

➢ Advanced plan generation system

Planning-Goal Stack Algorithm

One of the earliest techniques is planning using goal stack. Problem solver uses single stack that contains

- sub goals and operators both
- sub goals are solved linearly and then finally the conjoined sub goal is solved.

Plans generated by this method will contain complete sequence of operations for solving one goal followed by complete sequence of operations for the next etc.
Problem solver also relies on

- A database that describes the current situation.
- Set of operators with precondition, add and delete lists.

Let us assume that the goal to be satisfied is:

GOAL = G1 ^ G2 ^ … ^Gn

Sub-goals G1, G2, … Gn are stacked with compound goal G1 ^ G2 ^ …^ Gn at the bottom.

Top **G1**

G2

:

Gn

Bottom **G1 ^ G2 ^ … ^ G4**

At each step of problem solving process, the top goal on the stack is pursued

Algorithm

Find an operator that satisfies sub goal G1 (makes it true) and replace G1 by the operator.
If more than one operator satisfies the sub goal then apply some heuristic to choose one.
In order to execute the top most operation, its preconditions are added onto the stack.
Once preconditions of an operator are satisfied, then we are guaranteed that operator can be applied to produce a new state.
New state is obtained by using ADD and DELETE lists of an operator to the existing database.
Problem solver keeps tract of operators applied.
This process is continued till the goal stack is empty and problem solver returns the plan of the problem.

- **K STRIPS**

Modal Operator K :

We are familiar with the use of connectives ∧ and V in logics. Thinking of these connectives as operators that construct more complex formulas from simpler components. Here, we want to construct a formula whose intended meaning is that a certain agent knows a certain proposition.

The components consist of a term denoting the agent and a formula denoting a proposition that the agent knows. To accomplish this, modal operator K is introduced. For example, to say that Robot (name of agent) know that block A is on block B, then write,

K(Robot, On(A,B))

The sentence formed by combining K with the term Robot and the formula On(A,B) gets a new formula, the intended meaning of which is "Robot knows that block A is on block B".

The words "knows" and "belief" is different in meaning. That means an agent can believe a false proposition, but it cannot know anything that is false.

Some examples,

K(Agent1, K(Agent2, On(A,B))], means Agent1 knows that Agent1 knows that A is on B.

K(Agent1, On(A,B)) V K(Agent1, On(A,C)) means that either Agent1 knows that A is on B or it knows that A is on C.

K(Agent1, On(A,B)) V K(Agent1, ¬On(A,B)) means that either Agent1 knows whether or not A is on B.

Example in Planning Speech Action:

We can treat speech acts just like other agent systems. Our agent can use a plan-generating system to make plans comprising speech acts and other actions. To do so, it needs a model of the effects of these actions.

Consider for example, Tell(A, φ) , where A is Agent and φ is true.

We could model the effects of that action by the STRIPS rule :

Tell(A, φ) :

Precondition : Next_to(A) ∧ φ ∧ ¬K(A, φ)

Delete : ¬K(A, φ)

Add : K(A, φ)

- **Strategic explonation**

What is AI Strategy?

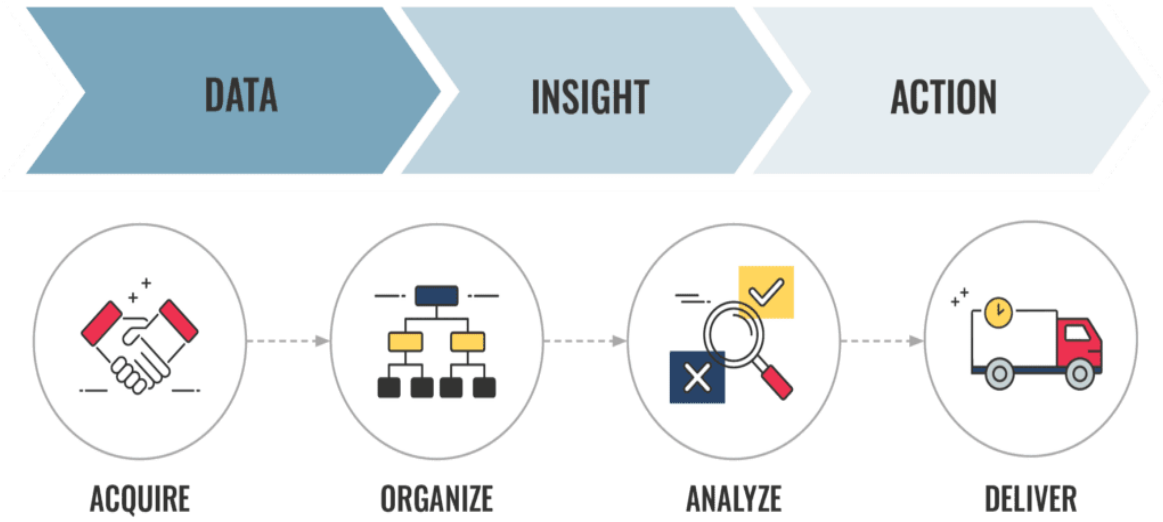
As your organization grows, it will produce more and more data. Just as having an effective data strategy will make sure that information growth can be properly managed, an effective AI strategy will make sure that that information growth translates to business value. Use your data to:

- Segment customers and products into groups that have similar behaviors and needs
- Predict customer purchases and churn risk
- Estimate the lifetime value of a customer or product
- Optimize manufacturing supply chains and perform predictive maintenance to increase uptime

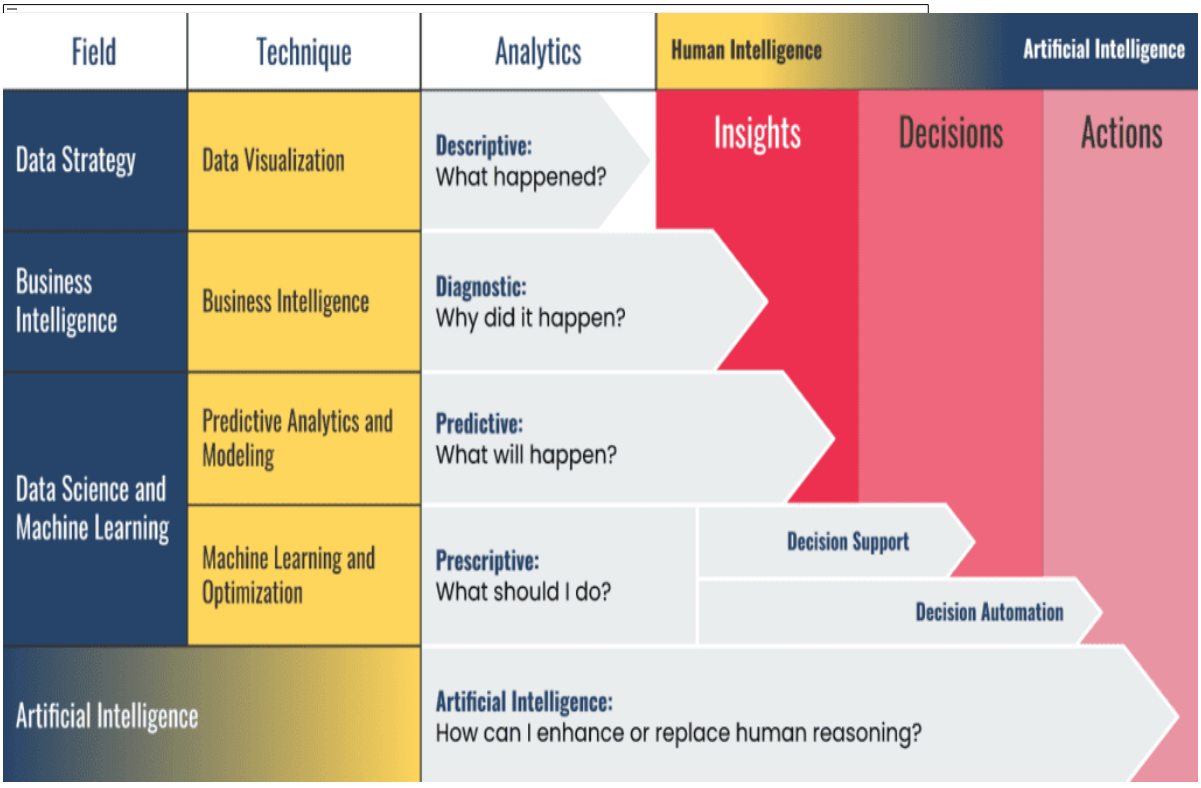
Without an effective strategy and roadmap, many companies find themselves at a technological dead end: the technologies they initially selected don't scale or support cutting-edge AI when it is developed. Bad strategy leads to siloed projects that don't build upon each other into a comprehensive AI program.

Effective AI strategies are opinionated and actionable. They are based on the real-life experiences of AI practitioners and deliver results.

With a time-tested AI strategy the investments you're making today will continue to provide value well into the future.



Why should you build an AI strategy?



What is (and is not) covered in an AI Strategy project?

No two companies deal with the same set of challenges, and an "out-of-the-box" AI strategy doesn't exist. Here are some examples of what is typically covered and not covered.

What is typically covered?

- Selecting the best technologies for your use cases
- Identifying governance requirements and compatible tools or processes
- Organizing your technologies into an effective tech stack
- Developing a phased implementation plan with realistic timeline and cost estimates
- Establishing training and upskilling guidelines
- Planning team structures and operational processes that scale as you grow

What is typically considered out of scope?

- Study of legal requirements related to model and data governance
- Specific data transformations and quality evaluation
- Auditing of information security practices and architectures

How our AI Strategy framework accelerates your project

We focus on efficient decision making and leveraging best practices to guide AI engagements. We make recommendations with the intent of showing immediate impact, and create a practical roadmap with achievable goals.

Quicker time to value
Services leverage a library of reference architectures and best practices to set up organization for success in 4-6 weeks.

Investment with immediate impact
Engagements typically cost between $40K and $100K and you'll be ready to implement the operating model and establish your AI program.

Proven best practices
Developed through working with the most advanced data and analytics organizations and deploying 1000s of AI use cases

Built by hands-on practitioners
Because marketing doesn't always translate to reality, our AI engagements are opinionated and involve Principal Solutions Architects who have worked on dozens of AI platforms and projects. We'll build a strategy tailored to your goals and capabilities (and we often help organizations execute on these strategies)

A.G. & S.G. Siddhartha Degree College of Arts & Science

Vuyyuru-521165, Krishna District, Andhra Pradesh

Value Added Course

Title: ARTIFICIAL INTELLIGENCE

Test Exercise:

1. An Artificial Intelligence system developed by Terry A. Winograd to permit an interactive dialogue about a domain he called blocks-world.

2. DARPA, the agency that has funded a great deal of American Artificial Intelligence research, is part of the Department of:

3. Who is the "father" of artificial intelligence?

4. KEE is a product of:

5. Default reasoning is another type of -

6. If a robot can alter its own trajectory in response to external conditions, it is considered to be:

7. One of the leading American robotics centers is the Robotics Institute located at

8. What is the name of the computer program that contains the distilled knowledge of an expert?

9. In LISP, the function evaluates both <variable> and <object> is -

10. The first ai programming language was called

A.G. & S.G. Siddhartha Degree College of Arts & Science

Vuyyuru-521165, Krishna District, Andhra Pradesh

Value Added Course

Title: ARTIFICIAL INTELLIGENCE

Key:

1. **BACON**
2. **DEFENSE**
3. **FISHER ADA**
4. **INTELLICORPN**
5. **NON-MONOTONIC REASONING**
6. **INTELLIGENT**
7. **CMU**
8. **EXPERT SYSTEM**
9. **SET**
10. **IPL**

A.G. & S.G. Siddhartha Degree College of Arts & Science
Vuyyuru-521165, Krishna District, Andhra Pradesh

Department of COMPUTER SCIENCE

Value Added Course
Title: ARTIFICIAL INTELLIGENCE

Marks List

Class: II MPC's

S. No	Roll No.	Name of the Student	Marks
1	20.601	Chillimuntha Lokesh	07
2	20.602	Loya Naga Padma	09
3	20.604	Pulipaka Chakradhar	10
4	20.605	Alaparthi Kumar Venkata Sai Kiran	07
5	20.606	Elusuri Poojitha	10
6	20.607	Rebba Bhanu Sree	10
7	20.608	Busi Manjusha	08
8	20.609	Gopalajoshula Prathyusha	10
9	20.610	Kolluri Likitha	07
10	20.611	Kurapati Harika	07
11	20.612	Goriparthi Sampoorna	10
12	20.613	Kolli Naga Tirumala Reddy	07
13	20.614	Namala V V Sandhya	08
14	20.615	Tumuluri Yogambika	10
15	20.616	Arilli Veeramma	10

16	20.617	Gangireddy Jahnavi	10
17	20.618	Katta Pravallika	08
18	20.619	Kagitha Hinduja	10
19	20.620	Ede Sravani Durga Devi	09
20	20.621	Munipalli Ajay	07
21	20.622	Dasari Naga Mallika	10
22	20.624	Juvvanapudi Sai Kiran	07
23	20.625	Taviti Ratna Kumari	10
24	20.626	Parimi Prem Chand	10
25	20.628	Shaik Sabiha Tasleem	10
26	20.629	Vasadi Radha	08
27	20.630	Chittibomma Poorna Chandra Moulika	08
28	20.631	Dokku Gnana Prasanna	09
29	20.633	Dabbada Sai	07
30	20.634	Mareedu Aliveni	10

Signature of Lecturer

Signature of HOD

PRINCIPAL

Signature of Principal

A.G. & S.G. Siddhartha Degree College of Arts & Science
Vuyyuru-521165, Krishna District, Andhra Pradesh

Department of COMPUTER SCIENCE

Value Added Course

Title: ARTIFICIAL INTELLIGENCE

Feed Back Form

1. Is the programme interested to you (Yes/No)
2. Have you attended all the session (Yes/No)
3. Is the content of the program is adequate (Yes/No)
4. Have the teacher covered the entire syllabus? (Yes/No)
5. Is the number of hours adequate? (Yes/No)
6. Do you have any suggestions for enhancing or reducing the number of weeks designed for the program? (Yes/No)
7. On the whole, is the program useful in terms of enriching your knowledge? (Yes/No)
8. Do you have any suggestions on the program? (Yes/No)

A.G. & S.G. Siddhartha Degree College of Arts & Science
Vuyyuru-521165, Krishna District, Andhra Pradesh

Value Added Course / Certificate Course - Attendance Register

Class / Section : II MPCs Year : II year Department of: Computer Science Paper: Artificial Intelligence Lecturer: G. Kalyayini

Sl.No	Roll No	Student Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total
1	20.601	Chillimuntha Lokesh	a	a	a	P	P	P	P	P	a	P	a	P	P	P	P	
2	20.602	Loya Naga Padma	P	P	P	P	a	P	P	P	P	P	P	P	P	a	P	
3	20.604	Pulipaka Chakradhar	P	P	P	P	P	P	P	a	a	a	P	a	P	a	P	
4	20.605	Alaparthi Kumar Venkata Sai Kiran	a	a	a	P	P	P	P	P	a	P	P	P	P	P	P	
5	20.606	Elusuri Poojitha	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
6	20.607	Rebba Bhanu Sree	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
7	20.608	Busi Manjusha	P	a	P	P	P	P	a	P	P	P	P	P	P	a	P	
8	20.609	Gopalajoshula Prathyusha	P	P	P	P	P	P	P	P	P	P	a	P	P	P	P	
9	20.610	Kolluri Likitha	a	P	P	P	a	P	P	P	P	P	P	P	P	P	a	
10	20.611	Kurapati Harika	P	P	P	P	P	a	P	P	P	P	P	a	P	P	P	
11	20.612	Goriparthi Sampoorna	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
12	20.613	Kolli Naga Tirumala Reddy	a	a	P	P	P	a	P	P	P	P	P	P	P	a	P	
13	20.614	Namala V V Sandhya	P	P	P	P	P	P	a	P	P	P	P	P	a	P	P	

14	20.615	Tumuluri Yogambika	P	P	P	a	P	P	P	P	P	P	P	a	P	P	
15	20.616	Arilli Veeramma	P	P	P	P	P	a	P	P	P	a	P	P	P	P	
16	20.617	Gangireddy Jahnavi	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
17	20.618	Katta Pravallika	P	P	P	P	P	P	a	P	P	P	P	P	P	P	
18	20.619	Kagitha Hinduja	P	P	P	P	P	P	P	a	P	P	P	P	P	P	
19	20.620	Ede Sravani Durga Devi	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
20	20.621	Munipalli Ajay	P	P	P	P	a	P	P	P	P	P	a	P	P	P	
21	20.622	Dasari Naga Mallika	a	P	P	P	P	P	P	P	P	P	P	P	P	P	
22	20.624	Juvvanapudi Sai Kiran	P	a	P	P	P	P	P	a	P	P	P	P	P	P	
23	20.625	Taviti Ratna Kumari	P	P	P	P	P	P	P	P	P	a	a	P	P	P	
24	20.626	Parimi Prem Chand	P	P	P	P	a	P	P	P	a	P	P	P	P	a	
25	20.628	Shaik Sabiha Tasleem	P	P	P	P	P	P	P	P	P	P	P	P	a	P	
26	20.629	Vasadi Radha	P	P	P	P	P	P	P	P	P	a	P	P	P	a	
27	20.630	Chittibomma Poorna Chandra Moulika	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
28	20.631	Dokku Gnana Prasanna	P	a	P	P	P	P	P	P	P	P	P	P	a	P	
29	20.633	Dabbada Sai	P	P	P	P	P	P	P	a	P	P	P	P	P	P	
30	20.634	Mareedu Aliveni	P	a	P	P	P	P	P	P	P	P	P	P	P	P	

Signature of Lecturer

Signature of HOD

Signature of Principal

PRINCIPAL
AG & SG Siddhartha Degree College of Arts&Science (Autonomous),Vuyyuru

A.G. & S.G. Siddhartha Degree College of Arts & Science

Vuyyuru-521165, Krishna District, Andhra Pradesh

Value Added Course / Certificate Course - Attendance Register

Class / Section: II MPC's Year : II year Department of: Computer Science Paper: Artificial Intelligence Lecturer: G. Kalpyini

Sl.No	Roll No	Student Name	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Total
1	20.601	Chillimuntha Lokesh	P	a	P	P	a	P	P	P	a	P	P	P	P	a	P	
2	20.602	Loya Naga Padma	P	P	P	P	P	P	P	P	P	P	a	P	P	P	P	
3	20.604	Pulipaka Chakradhar	a	P	P	P	P	P	P	a	a	a	P	P	P	P	P	
4	20.605	Alaparthi Kumar Venkata Sai Kiran	P	P	P	P	P	P	P	P	P	a	P	P	P	P	P	
5	20.606	Elusuri Poojitha	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
6	20.607	Rebba Bhanu Sree	P	P	P	P	P	P	P	P	a	P	P	P	P	P	P	
7	20.608	Busi Manjusha	P	P	P	P	P	P	P	P	P	P	a	P	P	a	P	
8	20.609	Gopalajoshula Prathyusha	P	a	a	P	P	P	P	P	P	P	P	P	a	P	P	
9	20.610	Kolluri Likitha	P	P	P	P	P	P	P	a	P	P	P	P	P	P	P	
10	20.611	Kurapati Harika	P	P	P	P	P	P	P	P	P	a	P	P	a	P	P	
11	20.612	Goriparthi Sampoorna	a	a	P	P	P	P	P	P	P	P	P	P	P	P	P	
12	20.613	Kolli Naga Tirumala Reddy	P	P	P	P	a	P	P	a	P	P	P	P	P	P	P	
13	20.614	Namala V V Sandhya	P	P	P	P	P	P	P	P	P	a	P	P	P	P	P	

14	20.615	Tumuluri Yogambika	P	P	P	P	P	a	P	P	a	P	P	P	P	P	P	
15	20.616	Arilli Veeramma	P	P	P	P	P	P	P	P	P	P	P	P	a	P	P	
16	20.617	Gangireddy Jahnavi	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
17	20.618	Katta Pravallika	P	P	P	P	P	P	a	P	P	P	P	P	P	P	P	
18	20.619	Kagitha Hinduja	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
19	20.620	Ede Sravani Durga Devi	P	P	P	P	a	P	P	P	P	P	P	P	a	P	P	P
20	20.621	Munipalli Ajay	P	P	a	P	P	P	P	a	P	P	P	a	P	P	P	
21	20.622	Dasari Naga Mallika	P	P	P	P	P	a	P	P	P	P	a	P	P	a	P	
22	20.624	Juvvanapudi Sai Kiran	P	P	P	P	P	a	P	P	P	a	P	P	a	P	P	
23	20.625	Taviti Ratna Kumari	P	P	P	P	a	P	P	P	a	P	P	P	a	P	P	
24	20.626	Parimi Prem Chand	P	P	P	P	P	a	P	P	P	P	a	P	P	P	P	
25	20.628	Shaik Sabiha Tasleem	P	P	P	P	P	P	a	P	P	a	P	P	P	P	P	
26	20.629	Vasadi Radha	P	P	P	P	a	P	P	P	a	P	P	P	P	P	P	
27	20.630	Chittibomma Poorna Chandra Moulika	P	P	a	P	P	P	a	P	P	P	P	P	a	P	P	
28	20.631	Dokku Gnana Prasanna	P	P	P	P	a	P	P	P	a	P	P	a	P	P	P	
29	20.633	Dabbada Sai	P	P	a	P	P	P	a	P	P	P	a	P	P	P	P	
30	20.634	Mareedu Aliveni	P	P	P	P	P	a	P	P	P	a	P	P	P	P	P	

Signature of Lecturer

Signature of HOD

Signature of principal

ADUSUMILLI GOPALAKRISHNAIAH & SUGARCANE GROWERS

SIDDHARTHA DEGREE COLLEGE OF ARTS & SCIENCE

Vuyyuru-521 165, Krishna District, Andhra Pradesh

An Autonomous College in the Jurisdiction of Krishna University

Accredited by NAAC with "A" Grade

DEPARTMENT OF COMPUTER SCIENCE

VALUE ADDED COURSE:ARTIFICIAL INTELLIGENCE

VAC CODE:AIVAC101

CERTIFICATE

This is to Certify that Son /Daughter of shri/Smt has Successfully completed value added course in ARTIFICIAL INTELLIGENCE Conducted by the Department of COMPUTER SCIENCE from 18-11-2021 to 12-01-2022 We wish him /her bright future

Co-ordinator **Head of Department** **Principal**