Advanced AI

Dr.Vaishnaw G.Kale
Associate Professor
Department of Computer Science
Engineering & Applications
D.Y.Patil International University, Pune

D Y PATIL INTERNATIONAL UNIVERSITY
AKURDI PUNE

About the Course

Name of the subject:Advanced AI
Course Code:MCA 301
Programme:Master in Computer Applications
Year of study:Second Year
Semester :III
Specialization:Artificial Intelligence and Data Science
Course Type:Specialization Course
Academic Year:2023-24

Syllabus

Module-III:Knowledge Representation and Reasoning

Propositional Logic: Propositional Theorem Proving, Effective Propositional Model Checking, Agents Based on Propositional Logic, First-Order Logic, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic, knowledge representation using propositional and predicate logic, comparison of propositional and predicate logic, reasoning and Types of reasoning

D Y PATIL INTERNATIONAL UNIVERSITY
AKURDI PUNE

Books

Sr.No.	Text Books	Name of the Author
1	“Artificial Intelligence: A Modern Approach”	Stuart Russell and Peter Norvig
2	“A First Course in Artificial Intelligence”	Deepak Khemani
3	“Artificial Intelligence”	Elaine Rich, Kevin Knight and Nair
4	“Deep Learning”	Ian Goodfellow The MIT Press
Sr.No.	**Reference Books**	**Name of the Author**
1	“Artificial Intelligence: A new Synthesis”	Nilsson Nils J
2	“Artificial Intelligence”	Patrick Henry Winston
3	“Computational Intelligence: An Introduction”	Andries P. Engelbrecht
4	“Artificial Intelligence: Concepts and Applications”	Dr. Lavika Goel

Module 03.Knowledge Representation and Reasoning

Logical Agent

Logical agents use logic to make decisions and act in the world. The two most common kinds of logics are:

- Propositional logic
- First-order logic

when thinking about logical agents, we imagine that the agent has a knowledge base (KB) that contains logical sentences that describe the state of the world you could think of a KB as a database designed for storing and retrieving logical sentences there are two main KB operations: TELL and ASK

- TELL puts a new fact into the KB
- ASK queries the KB about what is known; this usually involves inference, where the KB can determine that facts are true (or likely to be true) even though they have not been explicitly told them

Logical Agent

1) Propositional Logic(PL)

- Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions.
- A proposition is a declarative statement which is either true or false.
- It is a technique of knowledge representation in logical and mathematical form.

Examples

a) It is Sunday.

b) The Sun rises from West (False proposition)

c) 3+3= 7(False proposition)

d) 5 is a prime number.

D Y PATIL INTERNATIONAL UNIVERSITY
AKURDI PUNE

Logical Agent

1) **Propositional Logic(PL)**

- Propositional logic is also called Boolean logic as it works on 0 and 1.
- In propositional logic, we use symbolic variables to represent the logic, and we can use any symbol for a representing a proposition, such A, B, C, P, Q, R, etc.
- Propositions can be either true or false, but it cannot be both.
- Propositional logic consists of an object, relations or function, and logical connectives.
- These connectives are also called logical operators.
- The propositions and connectives are the basic elements of the propositional logic.
- Connectives can be said as a logical operator which connects two sentences.
- A proposition formula which is always true is called tautology, and it is also called a valid sentence.
- A proposition formula which is always false is called Contradiction.
- Statements which are questions, commands, or opinions are not propositions such as "Where is Rohini", "How are you", "What is your name", are not propositions.

Logical Agent

Syntax of Propositional Logic(PL)

- The syntax of propositional logic defines the allowable sentences for the knowledge representation. There are two types of Propositions:

a. Atomic Propositions: They are the simple propositions. It consists of a single proposition symbol. These are the sentences which must be either true or false.

Example

2+2 is 4, it is an atomic proposition as it is a **true** fact.

"The Sun is cold" is also a proposition as it is a **false** fact

b. Compound propositions: They are constructed by combining simpler or atomic propositions, using parentheses and logical connectives.

- Example

"It is raining today, and street is wet."

"Ankit is a doctor, and his clinic is in Mumbai."

Logical Agent

Logical Connectives in Propositional Logic(PL)

- Logical connectives are used to connect two simpler propositions or representing a sentence logically. We can create compound propositions with the help of logical connectives.
- There are mainly five connectives, which are given as follows:

1. Negation: A sentence such as ¬ P is called negation of P. A literal can be either Positive literal or negative literal.
2. Conjunction: A sentence which has ∧ connective such as, P ∧ Q is called a conjunction.
   Example: Rohan is intelligent and hardworking. It can be written as,
   P= Rohan is intelligent,
   Q= Rohan is hardworking. → P∧ Q.
3. Disjunction: A sentence which has ∨ connective, such as P ∨ Q. is called disjunction, where P and Q are the propositions.
   Example: "Ritika is a doctor or Engineer",
   Here P= Ritika is Doctor. Q= Ritika is Engineer, so we can write it as P ∨ Q.

Logical Agent

Logical Connectives in Propositional Logic(PL)

4) Implication: A sentence such as $P \rightarrow Q$, is called an implication. Implications are also known as if-then rules. It can be represented as

If it is raining, then the street is wet.

Let P= It is raining, and Q= Street is wet, so it is represented as $P \rightarrow Q$

5) Biconditional: A sentence such as $P \Leftrightarrow Q$ is a Biconditional sentence, example If I am breathing, then I am alive

P= I am breathing, Q= I am alive, it can be represented as $P \Leftrightarrow Q$.

Limitations

- We cannot represent relations like ALL, some, or none with propositional logic. Example:

a. All the girls are intelligent.

b. Some apples are sweet.

- Propositional logic has limited expressive power.
- In propositional logic, we cannot describe statements in terms of their properties or logical relationships

Logical Agent

First Order Logic(FOL)

- Unfortunately, in propositional logic (PL), we can only represent the facts, which are either true or false.
- PL is not sufficient to represent the complex sentences or natural language statements.
- The propositional logic has very limited expressive power.

Consider the following sentence, which we cannot represent using PL logic.

- "Some humans are intelligent", or
- "Sachin likes cricket."

To represent the above statements, PL logic is not sufficient, so we required some more powerful logic, such as first-order logic.

Logical Agent

First Order Logic(FOL)/Predicate Logic/First Order Predicate Logic

- First-order logic is another way of knowledge representation in artificial intelligence.
- It is an extension to propositional logic.
- FOL is sufficiently expressive to represent the natural language statements in a concise way.
- 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.
- First-order logic (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world:
- **Objects:** A, B, people, numbers, colors, wars, theories, squares, pits, wumpus
- **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
- **Function:** Father of, best friend, third inning of, end of ...

Logical Agent

First Order Logic(FOL)/Predicate Logic/First Order Predicate Logic

As a natural language, first-order logic also has two main parts: **Syntax and Semantics**

Syntax of FOL

The syntax of FOL determines which collection of symbols is a logical expression in first-order logic. The basic syntactic elements of first-order logic are symbols. We write statements in short-hand notation in FOL.

Atomic Sentences

- Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.
- We can represent atomic sentences as **Predicate (term1, term 2... term n)**.

Example**:** Ravi and Ajay are brothers: => Brothers (Ravi, Ajay)

Chinky is a cat: => cat (Chinky)

Logical Agent

First Order Logic(FOL)/Predicate Logic/First Order Predicate Logic

Syntax of FOL

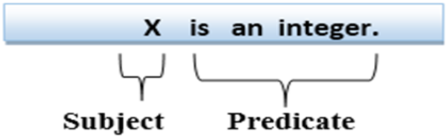
Complex Sentences

- Complex sentences are made by combining atomic sentences using connectives.

First-order logic statements can be divided into two parts:

- **Subject:** Subject is the main part of the statement.
- **Predicate:** A predicate can be defined as a relation, which binds two atoms together in a statement.

Consider the statement: "x is an integer", it consists of two parts, the first part x is the subject of the statement and second part "is an integer," is known as a predicate.



Logical Agent

Semantic of FOL

- An semantic of FOL assigns a notation to all symbols.
- It also determines a domain that specifies the range of quantifiers.
- Each term is assigned as an object, each predicate is assigned a property of objects and each sentence is assigned a true value.
- In this way FOL provides meaning to the terms, the predicates and formulas of the language.

Interpretation on FOL

- Domain D be an non empty set
- Each constant is assigned an element of D
- Each variable is assigned to subset of D
- Each function 'f' of 'm' is defined on m arguments of D and defines a mapping from D^m to D.

Logical Agent

Semantic of FOL

Expression 'E' and interpretation 'I' for 'E'

- The value of Truth symbol T and F
- The value of atomic sentence is either T or F determined by interpretation I
- The value of constant is the element of D, assigned by I
- Value of variables is set of elements of D, assigned by 'I'

The value of function expression is that element of D obtained by evaluating the function for parameter values assigned by I.

The value of negation of sentence is True if value of sentence is false and vice and versa.

The value of conjunction of two sentence is true if both are true otherwise false.

The true value of expression using are determined form the value of other operands

Knowledge based Agent

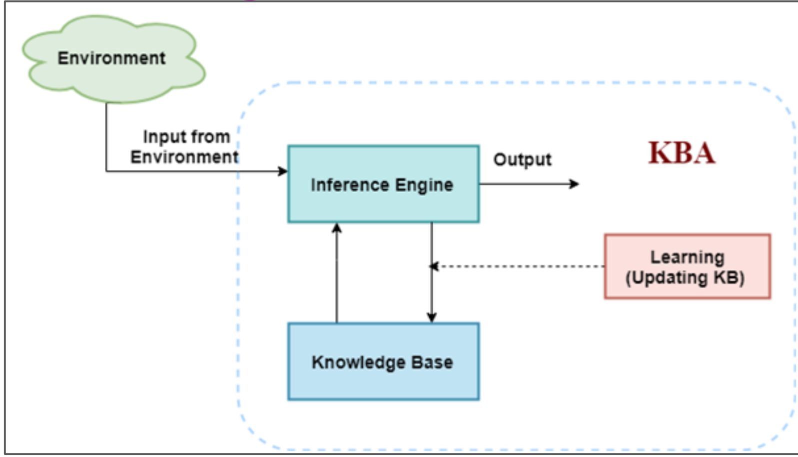
- An intelligent agent needs knowledge about the real world for taking decisions and reasoning to act efficiently.
- Knowledge-based agents are those agents who have the capability of maintaining an internal state of knowledge, reason over that knowledge, update their knowledge after observations and 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: domain specific content and
  - Inference system: Domain Independent system

Knowledge based Agent

D Y PATIL INTERNATIONAL UNIVERSITY AKURDI PUNE

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 above diagram is representing a generalized architecture for a knowledge-based agent.
- The knowledge-based agent (KBA) take 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 based Agent

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

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.

Knowledge based Agent

- 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.

D Y PATIL INTERNATIONAL UNIVERSITY AKURDI PUNE

Knowledge based Agent

Approaches

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

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. 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.

Propositional Theorem Proving

- We can show how entailment(the relationship between sentences whereby one sentence will be true if all the others are also true) can be done by theorem proving that is applying rules of inference directly to the sentences in our knowledge base to construct as proof of the desired sentence without consulting models.
- **Logical equivalence**: Two sentences Alpha and Beta are logically equivalent if they are true in the same set of models.

  we write this Alpha =Beta. For example we can easily show (using truth table) that P^Q and Q^P are logically equivalent.

- Tautological means Result should be true

Propositional Theorem Proving

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land\\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor\\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land\\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor\\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination}\\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition}\\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination}\\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination}\\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{De Morgan}\\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{De Morgan}\\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor\\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

Figure 7.11 Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

Propositional Theorem Proving

Sample Rules of Inference

- Modus Ponens: $\{\alpha \Rightarrow \beta, \alpha\} \vdash \beta$
- And Elimination: $\{\alpha \wedge \beta\} \vdash \alpha$; $\{\alpha \wedge \beta\} \vdash \beta$
- And Introduction: $\{\alpha, B\} \vdash \alpha \wedge \beta$
- Or introduction: $\{\alpha\} \vdash \alpha \vee \beta$
- Double negation Elimination: $\{\neg\neg\alpha\} \vdash \alpha$
- Implication Elimination: $\{\alpha \Rightarrow \beta\} \vdash \neg\alpha \vee \beta$
- Unit resolution: $\{\alpha \vee \beta, \neg\beta\} \vdash \alpha$
- Resolution: $\{\alpha \vee \beta, \neg\beta \vee \gamma\} \vdash \alpha \vee \gamma$

Sample Proof

If John is not married he is a bachelor. $(\neg P \Rightarrow Q)$
John is not a bachelor. $(\neg Q)$
Therefore, he is married. (P)

$\neg P \Rightarrow Q$

$\neg\neg P \vee Q$ — Implication elimination

$P \vee Q$, $\neg Q$ — Double negation elimination

P — Unit resolution

Could also check validity of:

$((\neg P \Rightarrow Q) \wedge \neg Q) \Rightarrow P$

Form of modus tollens reasoning

Agents based on Propositional Logic

- Agent need to deduce the word given percepts. we need complete logic models. we need agent to keep the track of the world. we need agent to make plans.
- The knowledge base is used by the agent for deducing what to do?.
- Knowledge base has Axioms(Axioms are sentences) and percept sentences from agent experiences.
- A part of the world that changes is said to be fluent(state variable).
- Permanent aspect without needing a time stamp are called Atemporal variables.

1) Assert the location fluent

2) Set of sentences to serve as transition model to track the fluent

3)Need prepositional symbols for actions indexed by time

4) Percept for time goes first and the action

Agents based on Propositional Logic

- We require sentences for each times, each direction and the same for the other actions like Grab, shoot, climb, turn left, turn right.
- Consider when the agent moves forward , this fact goes into KB.
- When we forgot to include what has remain unchanged, the need of this leads to frame problem.
- To resolve this problem we can add from axioms but this is really inefficient. This specific problem often is called as representational frame problem.
- To resolve this problem we can write axioms about fluent not actions.
- With successor state axioms and previously defined axioms the agent can ASK anything that can be answered in the world.
- Use of hybrid agent which combine condition action rules. This agent keeps an update plan and KB. KB is initialized with atemporal axioms.
- In this as time passes new percept axioms are added and all axioms related to time(Successor time axioms).
- The agent moves inferences by using ASK to determine safe moves.

Using First Order Logic

- Assertions and queries in FOL
- Adding sentences to KB using TELL called assertion
- Assertions: John is king, Richard is a person, and all kings are person

TELL(KB, King(John))
TELL(KB, Person(Richard))
TELL(KB, $\forall x$ King(x) $\Rightarrow$ Person(x))

- Ask questions of the KB using ASK

ASK(KB, King(John))

return true value

- Questions are called queries or goals. queries are logically entailed by KB should be answered affirmatively ASK(KB, Person(John)) should return true too

Using First Order Logic

- can ask quantified queries

ASK (KB, ∃x Person(x)) like asking for time and getting the answer true

- How about asking about what makes something true

ASKVARS(KB, Person(x)) Get string of answers

- Consider family relations

Example facts : E is a mother of C, C is the father of W, Ones grandmother is mother of one's parent

Objects in domain are people

- Two unary predicate: male, female
- Kinship relations are binary predicates :parents, siblings ,brother, sister, child, daughter, son , spouse, etc
- Functions of mother and father, each person has exactly one

Using First Order Logic

- Consider each function and predicate expressed in terms of other symbols

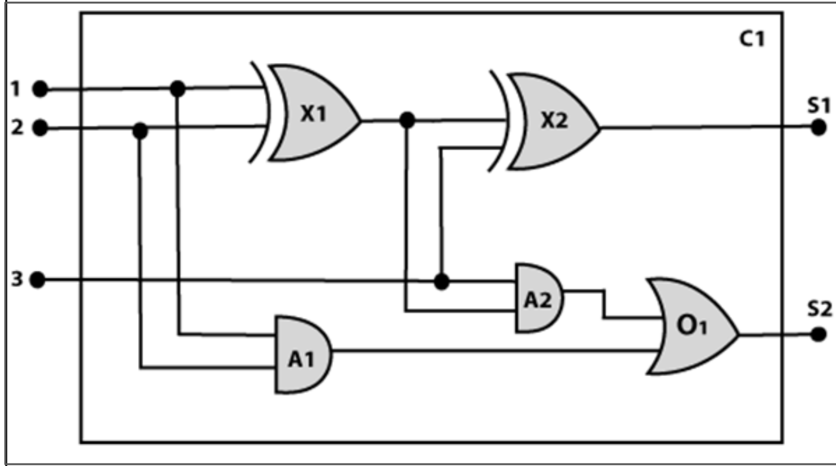
- One's mother is one's female parent:
  $\forall m,c\ Mother(c)=m \Leftrightarrow Female(m) \wedge Parent(m,c)$
- One's father is one's male spouse:
  $\forall w,h\ Husband(h,w) \Leftrightarrow Male(h) \wedge Spouse(h,w)$
- Male and female are disjoint categories:
  $\forall x\ Male(x) \Leftrightarrow \neg Female(x)$
- Parent and child are inverse relations:
  $\forall p,c\ Parent(p,c) \Leftrightarrow Child(c,p)$
- A grandparent is a parent of one's parent:
  $\forall g,c\ Grandparent(g,c) \Leftrightarrow \exists p\ Parent(g,p) \wedge Parent(p,c)$

The sentence are axioms in the domain where axiom are fundamental facts

- Provides basic factual information for deriving useful conclusions
- Also definitions, defining functions, and predicate in terms of other predicate
- Kind of like how software packages are built up by definitions of functions using library functions

Knowledge Engineering in FOL

- The process of constructing a knowledge-base in first-order logic is called as knowledge- engineering.
- The Knowledge Engineering Process
- Following are some main steps of the knowledge-engineering process. Using these steps, we will develop a knowledge base which will allow us to reason about digital circuit (**One-bit full adder**) which is shown here



Knowledge Engineering in FOL

1.Identify the task

The first step of the process is to identify the task, and for the digital circuit, there are various reasoning tasks.

At the first level or highest level, we will examine the functionality of the circuit:

- Does the circuit add properly?
- What will be the output of gate A2, if all the inputs are high?

At the second level, we will examine the circuit structure details such as:

- Which gate is connected to the first input terminal?
- Does the circuit have feedback loops?

Knowledge Engineering in FOL

2.Assemble the relevant knowledge

In the second step, we will assemble the relevant knowledge which is required for digital circuits. So for digital circuits, we have the following required knowledge:

- Logic circuits are made up of wires and gates.
- Signal flows through wires to the input terminal of the gate, and each gate produces the corresponding output which flows further.
- In this logic circuit, there are four types of gates used: **AND, OR, XOR, and NOT**.
- All these gates have one output terminal and two input terminals (except NOT gate, it has one input terminal).

Knowledge Engineering in FOL

3.Decide on Vocabulary

- The next step of the process is to select functions, predicate, and constants to represent the circuits, terminals, signals, and gates.
- Firstly we will distinguish the gates from each other and from other objects.
- Each gate is represented as an object which is named by a constant, such as, **Gate(X1)**.
- The functionality of each gate is determined by its type, which is taken as constants such as **AND, OR, XOR, or NOT**.
- Circuits will be identified by a predicate: **Circuit (C1)**.
- For the terminal, we will use predicate: **Terminal(x)**.
- For gate input, we will use the function **In(1, X1)** for denoting the first input terminal of the gate, and for output terminal we will use **Out (1, X1)**.
- The function **Arity(c, i, j)** is used to denote that circuit c has i input, j output.
- The connectivity between gates can be represented by predicate **Connect(Out(1, X1), In(1, X1))**.
- We use a unary predicate **On (t)**, which is true if the signal at a terminal is on.

Knowledge Engineering in FOL

4. Encode general knowledge of the domain

5.Encode a description of the problem instance:

- Now we encode problem of circuit C1, firstly we categorize the circuit and its gate components.
- This step is easy if ontology about the problem is already thought.
- This step involves the writing simple atomics sentences of instances of concepts, which is known as ontology.
- For the given circuit C1, we can encode the problem instance in atomic sentences as below:
- Since in the circuit there are two XOR, two AND, and one OR gate so atomic sentences for these gates will be:

1. For XOR gate: Type(x1)= XOR, Type(X2) = XOR
2. For AND gate: Type(A1) = AND, Type(A2)= AND
3. For OR gate: Type (O1) = OR.

And then represent the connections between all the gates.

Knowledge Engineering in FOL

6. Pose queries to the inference procedure and get answers:

- In this step, we will find all the possible set of values of all the terminal for the adder circuit.
- The first query will be: What should be the combination of input which would generate the first output of circuit C1, as 0 and a second output to be 1?

7. Debug the knowledge base:

- Now we will debug the knowledge base, and this is the last step of the complete process.
- In this step, we will try to debug the issues of knowledge base.
- In the knowledge base, we may have omitted assertions like $1 \neq 0$.

Difference between Propositional & Predicate Logic

1. Propositional Logic :

A **proposition** is basically a declarative sentence that has a truth value.

- Truth value can either be true or false, but it needs to be assigned any of the two values and not be ambiguous.
- The purpose of using propositional logic is to analyze a statement, individually or compositely.

For example :

The following statements :

1. If x is real, then x2 > 0
2. What is your name?
3. (a+b)2 = 100
4. This statement is true.

Are not propositions because they do not have a truth value. They are ambiguous.

Difference between Propositional & Predicate Logic

- But the following statements :

1. (a+b)2 = a2 + 2ab + b2
2. If x is real, then x2 >= 0
3. If x is real, then x2 < 0
4. The sun rises in the east.
5. The sun rises in the west.

- Are all propositions because they have a specific truth value, true or false.
- The branch of logic that deals with proposition is **propositional logic**.

Difference between Propositional & Predicate Logic

2. Predicate Logic :

- Predicates are properties, additional information to better express the subject of the sentence.
- A quantified predicate is a proposition , that is, when you assign values to a predicate with variables it can be made a proposition.

For example :

In P(x) : x>5, x is the subject or the variable and '>5' is the predicate.

P(7) : 7>5 is a proposition where we are assigning values to the variable x, and it has a truth value, i.e. True.

Difference between Propositional & Predicate Logic

Propositional Logic	Predicate Logic
Deals with a collection of declarative statements which have a truth value, true or false.	It is an expression consisting of variables with a specified domain. It consists of objects, relations and functions between the objects.
It is the basic and most widely used logic. Also known as Boolean logic.	It is an extension of propositional logic covering predicates and quantification.
A proposition has a specific truth value, either true or false.	A predicate truth value depends on the variables' value.
Scope analysis is not done in propositional logic.	Predicate logic helps analyze the scope of the subject over the predicate.

Difference between Propositional & Predicate Logic

Propositional Logic	Predicate Logic
Propositions are combined with Logical Operators or Logical Connectives like Negation(¬), Disjunction(∨), Conjunction(∧), Exclusive OR(⊕), Implication(⇒), Bi-Conditional or Double Implication(⇔).	Predicate Logic adds by introducing quantifiers to the existing proposition.
It is a more generalized representation.	It is a more specialized representation.
It cannot deal with sets of entities.	It can deal with set of entities with the help of quantifiers.

Reasoning

- The reasoning is the mental process of deriving logical conclusion 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.
- In artificial intelligence, the reasoning is essential so that the machine can also think rationally as a human brain, and can perform like a human.

Types of Reasoning

In artificial intelligence, reasoning can be divided into the following categories:

  - Deductive reasoning, Inductive reasoning, Abductive reasoning, Common Sense Reasoning,Monotonic Reasoning,Non-monotonic Reasoning.

Reasoning

1) Deductive Reasoning

- Deductive reasoning is deducing new information from logically related known information. It is the form of valid reasoning, which means the argument's conclusion must be true when the premises are true.
- Deductive reasoning is a type of propositional logic in AI, and it requires various rules and facts.
- It is sometimes referred to as top-down reasoning, and contradictory to inductive reasoning.
- In deductive reasoning, the truth of the premises guarantees the truth of the conclusion.
- Deductive reasoning mostly starts from the general premises to the specific conclusion

Reasoning

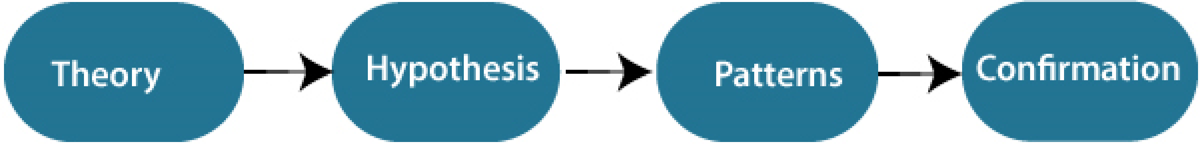
Example:

Premise-1: All the human eats veggies

Premise-2: Suresh is human.

Conclusion: Suresh eats veggies.

The general process of deductive reasoning is given below:



Reasoning

2) Inductive Reasoning

- Inductive reasoning is a form of reasoning to arrive at a conclusion using limited sets of facts by the process of generalization. It starts with the series of specific facts or data and reaches to a general statement or conclusion.
- Inductive reasoning is a type of propositional logic, which is also known as cause-effect reasoning or bottom-up reasoning.
- In inductive reasoning, we use historical data or various premises to generate a generic rule, for which premises support the conclusion.
- In inductive reasoning, premises provide probable supports to the conclusion, so the truth of premises does not guarantee the truth of the conclusion.

Reasoning

2) Inductive Reasoning

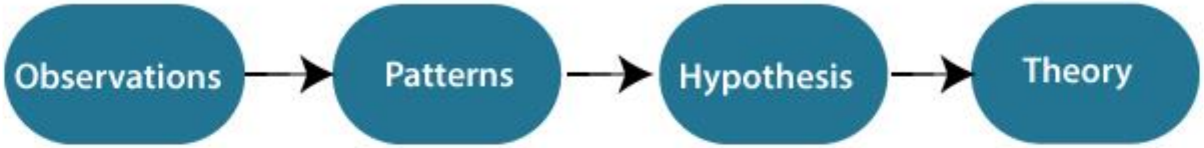
- Inductive reasoning is a form of reasoning to arrive at a conclusion using limited sets of facts by the process of generalization. It starts with the series of specific facts or data and reaches to a general statement or conclusion.
- Inductive reasoning is a type of propositional logic, which is also known as cause-effect reasoning or bottom-up reasoning.
- In inductive reasoning, we use historical data or various premises to generate a generic rule, for which premises support the conclusion.
- In inductive reasoning, premises provide probable supports to the conclusion, so the truth of premises does not guarantee the truth of the conclusion.

Reasoning

Example:

Premise: All of the pigeons we have seen in the zoo are white.

Conclusion: Therefore, we can expect all the pigeons to be white.



Reasoning

3) Abductive Reasoning

Abductive reasoning is a form of logical reasoning which starts with single or multiple observations then seeks to find the most likely explanation or conclusion for the observation.

Abductive reasoning is an extension of deductive reasoning, but in abductive reasoning, the premises do not guarantee the conclusion.

Example:

Implication: Cricket ground is wet if it is raining

Axiom: Cricket ground is wet.

Conclusion It is raining.

Reasoning

4) Common Sense Reasoning

- Common sense reasoning is an informal form of reasoning, which can be gained through experiences.
- Common Sense reasoning simulates the human ability to make presumptions about events which occurs on every day.
- It relies on good judgment rather than exact logic and operates on **heuristic knowledge** and **heuristic rules**.

Example:

1. One person can be at one place at a time.
2. If I put my hand in a fire, then it will burn.

- The above two statements are the examples of common sense reasoning which a human mind can easily understand and assume.

D Y PATIL INTERNATIONAL UNIVERSITY
AKURDI PUNE

Reasoning

5) Monotonic Reasoning

- In monotonic reasoning, once the conclusion is taken, then it will remain the same even if we add some other information to existing information in our knowledge base. In monotonic reasoning, adding knowledge does not decrease the set of prepositions that can be derived.
- To solve monotonic problems, we can derive the valid conclusion from the available facts only, and it will not be affected by new facts.
- Monotonic reasoning is not useful for the real-time systems, as in real time, facts get changed, so we cannot use monotonic reasoning.
- Monotonic reasoning is used in conventional reasoning systems, and a logic-based system is monotonic.
- Any theorem proving is an example of monotonic reasoning.

Reasoning

Example:

- **Earth revolves around the Sun.**

It is a true fact, and it cannot be changed even if we add another sentence in knowledge base like, "The moon revolves around the earth" Or "Earth is not round," etc.

Advantages of Monotonic Reasoning:

- In monotonic reasoning, each old proof will always remain valid.
- If we deduce some facts from available facts, then it will remain valid for always.

Disadvantages of Monotonic Reasoning:

- We cannot represent the real world scenarios using Monotonic reasoning.
- Hypothesis knowledge cannot be expressed with monotonic reasoning, which means facts should be true.
- so new knowledge from the real world cannot be added.

Reasoning

6) Non-monotonic Reasoning

- In Non-monotonic reasoning, some conclusions may be invalidated if we add some more information to our knowledge base.
- Logic will be said as non-monotonic if some conclusions can be invalidated by adding more knowledge into our knowledge base.
- Non-monotonic reasoning deals with incomplete and uncertain models.
- "Human perceptions for various things in daily life, "is a general example of non-monotonic reasoning.

D Y PATIL INTERNATIONAL UNIVERSITY
AKURDI PUNE

Reasoning

Example: Let suppose the knowledge base contains the following knowledge:

- **Birds can fly**
- **Penguins cannot fly**
- **Pitty is a bird**

So from the above sentences, we can conclude that **Pitty can fly**.

However, if we add one another sentence into knowledge base "**Pitty is a penguin**", which concludes "**Pitty cannot fly**", so it invalidates the above conclusion.

Reasoning

Advantages of Non-monotonic reasoning:

- For real-world systems such as Robot navigation, we can use non-monotonic reasoning.
- In Non-monotonic reasoning, we can choose probabilistic facts or can make assumptions.

Disadvantages of Non-monotonic Reasoning:

- In non-monotonic reasoning, the old facts may be invalidated by adding new sentences.
- It cannot be used for theorem **proving**.

Reasoning

- Consider the following two examples to better understand how reasoning works in artificial intelligence applications and systems:

1. Alexa: is a cognitive virtual assistant that employs reasoning to make recommendations and suggestions based on orders. For instance, the closest place, the date for tomorrow, the AM and PM, and so on.

2. WolframAlpha: To do mathematical computations based on meal portions, this computational knowledge engine uses reasoning.

- In brief, machines, like humans, use reasoning, knowledge representation, logic, and learning to analyze, solve problems, draw conclusions, and more.

Thank you!

D Y PATIL
INTERNATIONAL
UNIVERSITY
AKURDI PUNE