CAP 378
ARTIFICIAL INTELIGENCE

Tanzeela Javid Kaloo (32638)
Assistant Professor
System And Architecture
Lovely Professional University

UNIT – III
Knowledge Representation

Tanzeela Javid Kaloo (32638)

Assistant Professor

System And Architecture

Lovely Professional University

Content

- Types Of Knowledge In AI,
- Issues In Knowledge Representation,
- Logic Representation,
- Propositional Logic,
- Predicate Logic,
- Forward Chaining And Backward Chaining

Knowledge Representation

- **What to Represent?**

Let us first consider what kinds of knowledge might need to be represented in AI systems:

- **Objects**
  - -- Facts about objects in our world domain. *e.g.* Guitars have strings, trumpets are brass instruments.
- **Events**
  - -- Actions that occur in our world. *e.g.* Steve Vai played the guitar in Frank Zappa's Band.

- **Performance**
  - -- A behavior like *playing the guitar* involves knowledge about how to do things.
- **Meta-knowledge**
  - -- knowledge about what we know.

Thus in solving problems in AI we must represent knowledge and there are two entities to deal with:

- **Facts**
  - -- truths about the real world and what we represent. This can be regarded as the *knowledge level*
- **Representation of the facts**
  - which we manipulate. This can be regarded as the *symbol level* since we usually define the representation in terms of symbols that can be manipulated by programs.

- We can structure these entities at two levels
- **the knowledge level**
  - -- at which facts are described
- **the symbol level**
  - -- at which representations of objects are defined in terms of symbols that can be manipulated in programs.

- English or natural language is an obvious way of representing and handling facts.
- Logic enables us to consider the following fact:
- *spot is a dog* as *dog(spot)* We could then infer that all dogs have tails with:

  Vx : *dog(x)* → *hasatail(x)* We can then deduce:

  hasatail(Spot)

Using Knowledge (1)

We have briefly mentioned where knowledge is used in AI systems. Let us consider a little further to what applications and how knowledge may be used.

- **Learning** -- acquiring knowledge. This is more than simply adding new facts to a knowledge base.
- New data may have to be *classified* prior to storage for easy *retrieval, etc.*
- *Duplication should be avoided.*

- **Reasoning** -- Infer facts from existing data.

Using Knowledge(2)

If a system only knows:

- Davis is a Jazz Musician.
- All Jazz Musicians can play their instruments well.
- If things like *Is Davis a Jazz Musician?* or *Can Jazz Musicians play their instruments well?* are asked then the answer is readily obtained from the data structures and procedures.
- However a question like *Can Davis play his instrument well?* requires reasoning.

Properties for Knowledge Representation Systems

The following properties should be possessed by a knowledge representation system.

- **Representational Adequacy**
  - -- the ability to represent the required knowledge;
- **Inferential Adequacy**
  - - the ability to manipulate the knowledge represented to produce new knowledge corresponding to that inferred from the original;
- **Inferential Efficiency**
  - - the ability to direct the inferential mechanisms into the most productive directions by storing appropriate guides;
- **Acquisitional Efficiency**
  - - the ability to acquire new knowledge using automatic methods wherever possible rather than reliance on human intervention.

To date no single system optimises all of the above

Knowledge Representation

Knowledge in AI is represented using the Formal Language. Following are the logical languages used to represent the knowledge:

- Propositional Logic
- Predicate Logic
- First Order Logic
- Second Order Logic

Propositional Calculus

- Proposition is a declarative statement declaring some fact.
- It is either true or false but not both.
- Example
  - It is Raining
  - 2+2=6
  - Apples are black.
  - Two and two makes 5.
  - 2016 will be the lead year.
  - Delhi is in India.

- Propositional Symbols: P, Q, R, S, …
- Truth Symbols: true, false

Type of Propositions

In propositional logic, there are two types of propositions-

1. Atomic Proposition
2. Compound Proposition

Atomic Proposition

Atomic propositions are those propositions that can not be divided further. Small letters like p, q, r, s etc are used to represent atomic propositions.

Examples-

The examples of atomic propositions are-

p : Sun rises in the east.
q : Sun sets in the west.
r : Apples are red.
s : Grapes are green.

Compound Proposition

Compound propositions are those propositions that are formed by combining one or more atomic propositions using connectives.

In other words, compound propositions are those propositions that contain some connective.

Capital letters like P, Q, R, S etc are used to represent compound propositions.

Examples-

P : Sun rises in the east and Sun sets in the west.

Q : Apples are red and Grapes are green.

Statements That Are Not Propositions

Following kinds of statements are not propositions-

1. Command
2. Question
3. Exclamation
4. Inconsistent
5. Predicate or Proposition Function

Examples-

- Following statements are not propositions-

- Close the door. (Command)
- Do you speak French? (Question)
- What a beautiful picture! (Exclamation)
- I always tell lie. (Inconsistent)
- $P(x) : x + 3 = 5$ (Predicate)

Logical Connectives

Name of Connective	Connective Word	Symbol
Negation	Not	$\rceil$, ~, ‘, −
Conjunction	And	$\wedge$
Disjunction	Or	$\vee$
Conditional	If-then	$\rightarrow$
Biconditional	If and only if	$\leftrightarrow$

Logical Connectives

- **Negation** (¬): The negation of a proposition P is denoted as ¬P, meaning "not P".
  - Example: ¬P (if P is true, ¬P is false).
- **Conjunction** (∧): Represents "and". P∧Q is true if both P and Q are true.
  - Example: P∧Q (both P and Q must be true).
- **Disjunction** (∨): Represents "or". P∨Q is true if at least one of P or Q is true.
  - Example: P∨Q (either P or Q or both are true).
- **Implication** (→): Represents "if... then...". P→Q is false only when P is true and Q is false.
  - Example: "If it rains, then the ground will be wet" (P→Q).
- **Biconditional** (↔): Represents "if and only if". P↔Q is true when both P and Q are either both true or both false.
  - Example: "You will get an A if and only if you study" (P↔Q).

Negation

p	~p
F	T
T	F

Conjunction

p	q	p ∧ q
F	F	F
F	T	F
T	F	F
T	T	T

Disjunction

p	q	p ∨ q
F	F	F
F	T	T
T	F	T
T	T	T

Conditional

p	q	$p \rightarrow q$
F	F	T
F	T	T
T	F	F
T	T	T

Biconditional

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Propositional Calculus Sentences

- Every propositional and truth symbol is a sentence.
- Negation ($\neg P$), conjunction ($P \land Q$), disjunction ($P \lor Q$), implication ($P \to Q$), and equivalence ($P \lor Q \equiv R$) are all valid sentences.
- Sentences are also called well-formed formulas (WFFs).
- Examples of components:
  - $P \land Q$: Conjuncts are P and Q.
  - $P \to Q$: Premise is P, conclusion is Q.
- Parentheses and brackets control the order of evaluation (e.g., $(P \lor Q) \equiv R$ vs $P \lor (Q \equiv R)$).

Truth Assignment Rules

- **Negation (¬P)**: True if P is false, false if P is true.
- **Conjunction (P ∧ Q)**: True if both P and Q are true; otherwise, false.
- **Disjunction (P ∨ Q)**: False only when both P and Q are false; otherwise, true.
- **Implication (P → Q)**: False only when P is true and Q is false; otherwise, true.
- **Equivalence (P ≡ Q)**: True if both expressions have the same truth value in all interpretations.

Logical Identities

- **Examples of Logical Identities**:
  - $\neg(\neg P) \equiv P$ (Double negation)
  - $P \rightarrow Q \equiv \neg P \vee Q$ (implication is equivalent to "not P or Q").
  - $P \rightarrow Q \equiv \neg Q \rightarrow \neg P$ (contrapositive law)
  - $(\neg P \rightarrow Q) \equiv (P \vee Q)$ (Contrapositive law)
  - de Morgan's Laws: $\neg (P \vee Q) \equiv (\neg P \wedge \neg Q)$, $\neg (P \wedge Q) \equiv (\neg P \vee \neg Q)$
  - Commutative: $(P \wedge Q) \equiv (Q \wedge P)$
  - Associative: $((P \wedge Q) \wedge R) \equiv (P \wedge (Q \wedge R))$
  - Distributive: $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Focuses on *what* is true or logically consistent.

Consistency: A set of propositions is **consistent** if there is at least one interpretation where all the propositions are true

Rules

- **Rules of Inference**: Including rules like
  - **Modus Ponens:** From $P \rightarrow Q$ and P, infer Q (if $P \rightarrow Q$ and P then Q) and
  - **Modus Tollens:** $P \rightarrow Q$ and $\neg Q$, infer $\neg P$. (if $P \rightarrow Q$ and $\neg Q$ then $\neg P$).
  - **Disjunctive Syllogism**: From $P \vee Q$ and $\neg P$, infer Q.
  - **Hypothetical Syllogism**: From $P \rightarrow Q$ and $Q \rightarrow R$, infer $P \rightarrow R$.

Validity of Arguments

- An **argument** in propositional logic consists of premises and a conclusion.
- The argument is **valid** if, whenever all the premises are true, the conclusion must be true.

Limitations of Propositional Logic

- **Static Nature**:
  - Propositional logic requires explicit enumeration of all relationships.
  - Dynamic relationships (e.g., "Find the nth ancestor") are cumbersome.
- **No Variables for Quantifiers**:
  - Statements like "All descendants of Peter" require predicate logic, not propositional logic.
- **Limited Expressiveness**:
  - Cannot generalize rules for complex reasoning (e.g., inheritance of traits or roles).

Converting English sentences to Propositional Logic

Word	Replacement
And	Conjunction (∧)
Or	Disjunction (∨)
But	And
Whenever	If
When	If
Either p or q	p or q
Neither p nor q	Not p and Not q
p unless q	$\sim q \rightarrow p$
p is necessary but not sufficient for q	$(q \rightarrow p)$ and $\sim(p \rightarrow q)$

Examples

Q: Write the following English sentences in symbolic form-

1. If it rains, then I will stay at home.
2. If I will go to Australia, then I will earn more money.
3. He is poor but honest.
4. If a = b and b = c then a = c.
5. Neither it is hot nor cold today.
6. He goes to play a match if and only if it does not rain.
7. Birds fly if and only if sky is clear.
8. I will go only if he stays.
9. I will go if he stays.
10. It is false that he is poor but not honest.

Solution:

If it rains, then I will stay at home.

The given sentence is- “If it rains, then I will stay at home.”
This sentence is of the form- “If p then q”.

So, the symbolic form is $\mathbf{p \rightarrow q}$ where-

p : It rains

q : I will stay at home

Solution:

He is poor but honest.

The given sentence is- “He is poor but honest.”
We can replace “but” with “and”.
Then, the sentence is- “He is poor and honest.”

So, the symbolic form is $\mathbf{p \wedge q}$ where-

p : He is poor

q : He is honest

Solution:

If a = b and b = c then a = c.

The given sentence is- “If a = b and b = c then a = c.”
This sentence is of the form- “If p then q”.

So, the symbolic form is **$(p \wedge q) \rightarrow r$** where-

p : a = b

q : b = c

r : a = c

Solution:
Neither it is hot nor cold today.

The given sentence is- “Neither it is hot nor cold today.”
This sentence is of the form- “Neither p nor q”.
“Neither p nor q” can be re-written as “Not p and Not q”.

So, the symbolic form is $\mathbf{\sim p \land \sim q}$ where-

p : It is hot today

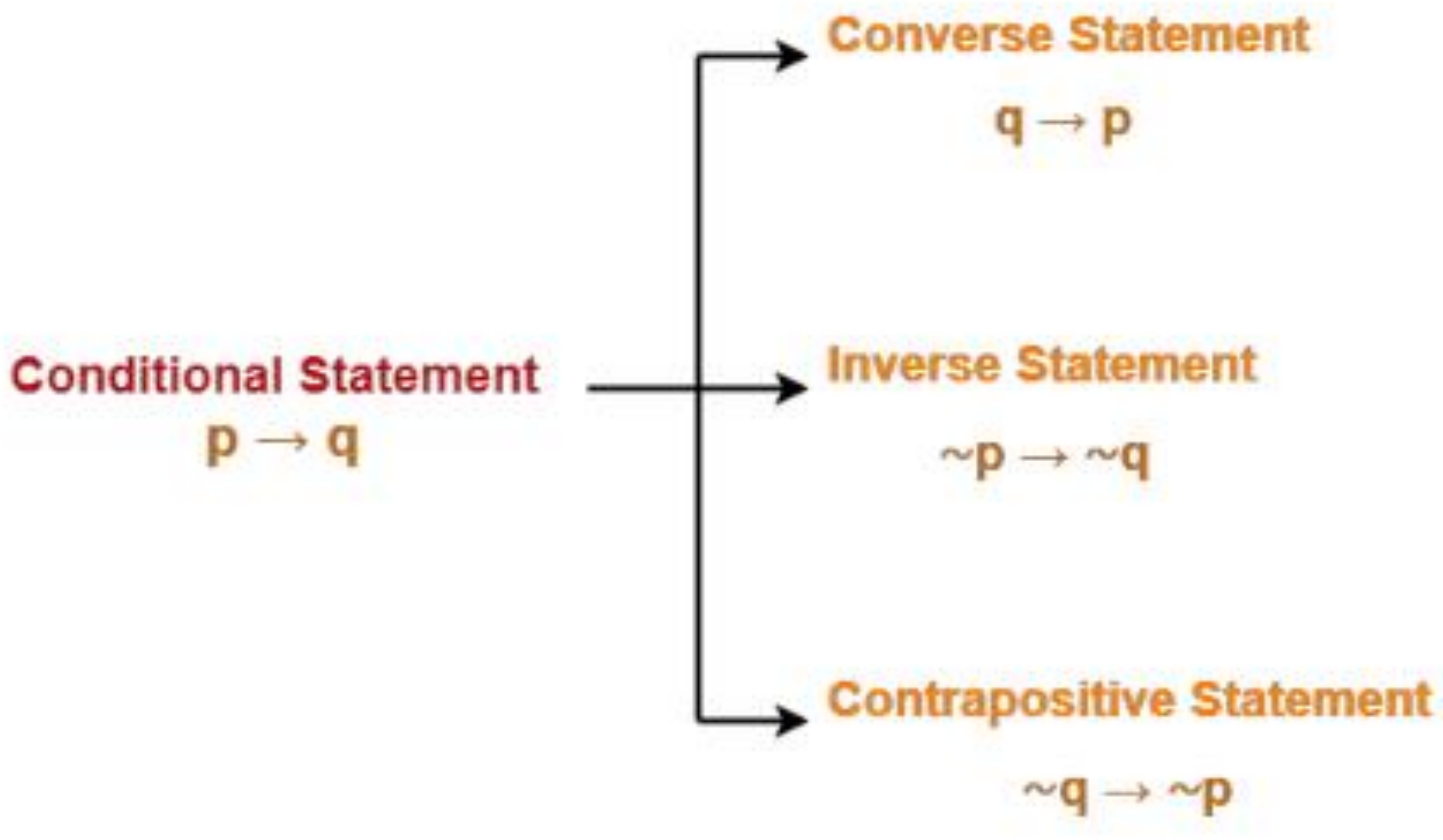
q : It is cold today

Examples

Q: Write the following English sentences in symbolic form-

1. It is false that he is poor or clever but not honest.
2. It is hot or else it is both cold and cloudy.
3. I will not go to class unless you come.
4. We will leave whenever he comes.
5. Either today is Sunday or Monday.
6. You will qualify GATE only if you work hard.
7. Presence of cycle in a single instance RAG is a necessary and sufficient condition for deadlock.
8. Presence of cycle in a multi-instance RAG is a necessary but not sufficient condition for deadlock.
9. I will dance only if you sing.
10. Neither the red nor the green is available in size 5.

Converse, Inverse and Contrapositive



Important Notes-

Note-01:

- For conditional statements ($p \rightarrow q$) only, the converse, inverse and contrapositive statements can be written.

Note-02:

Performing any two actions always result in the third one.

For example-

- Inverse of converse is contrapositive.
- Inverse of contrapositive is converse.
- Converse of inverse is contrapositive.
- Converse of contrapositive is inverse.
- Contrapositive of inverse is converse.
- Contrapositive of converse is inverse.

Note-03:

For a conditional statement $p \rightarrow q$,

- Its converse statement ($q \rightarrow p$) and inverse statement ($\sim p \rightarrow \sim q$) are equivalent to each other.
- $p \rightarrow q$ and its contrapositive statement ($\sim q \rightarrow \sim p$) are equivalent to each other.

Write the converse, inverse and contrapositive of the following statements-

1. If today is Sunday, then it is a holiday.
2. If $5x – 1 = 9$, then $x = 2$.
3. If it rains, then I will stay at home.
4. I will dance only if you sing.
5. I will go if he stays.
6. We leave whenever he comes.
7. You will qualify GATE only if you work hard.
8. If you are intelligent, then you will pass the exam.

- The given sentence is- “If today is Sunday, then it is a holiday.”
- This sentence is of the form- “If p then q”.

So, the symbolic form is $\mathbf{p} \rightarrow \mathbf{q}$ where-

p : Today is Sunday

q : It is a holiday

Converse Statement- If it is a holiday, then today is Sunday.

Inverse Statement- If today is not Sunday, then it is not a holiday.

Contrapositive Statement- If it is not a holiday, then today is not Sunday.

- The given sentence is- “If $5x – 1 = 9$, then $x = 2$.”
- This sentence is of the form- “If p then q”.

So, the symbolic form is $\mathbf{p} \rightarrow \mathbf{q}$ where-

$p : 5x – 1 = 9$

$q : x = 2$

Converse Statement- If $x = 2$, then $5x – 1 = 9$.

Inverse Statement- If $5x – 1 \neq 9$, then $x \neq 2$.

Contrapositive Statement- If $x \neq 2$, then $5x – 1 \neq 9$.

- The given sentence is- “If it rains, then I will stay at home.”
- This sentence is of the form- “If p then q”.

So, the symbolic form is **p → q** where-

p : It rains

q : I will stay at home

Converse Statement- If I will stay at home, then it rains.

Inverse Statement- If it does not rain, then I will not stay at home.

Contrapositive Statement- If I will not stay at home, then it does not rain.

- The given sentence is- “I will dance only if you sing.”
- This sentence is of the form- “p only if q”.

So, the symbolic form is $\mathbf{p} \rightarrow \mathbf{q}$ where-

p : I will dance

q : You sing

Converse Statement- If you sing, then I will dance.

Inverse Statement- If I will not dance, then you do not sing.

Contrapositive Statement- If you do not sing, then I will not dance.

- The given sentence is- “I will go if he stays.”
- This sentence is of the form- “q if p”.

So, the symbolic form is $\mathbf{p \rightarrow q}$ where-

p : He stays

q : I will go

Converse Statement- If I will go, then he stays.

Inverse Statement- If he does not stay, then I will not go.

Contrapositive Statement- If I will not go, then he does not stay.

- The given sentence is- “We leave whenever he comes.”
- We can replace “whenever” with “if”.
- Then, the sentence is- “We leave if he comes.”
- This sentence is of the form- “q if p”.

So, the symbolic form is $\mathbf{p} \rightarrow \mathbf{q}$ where-

p : He comes

q : We leave

Converse Statement- If we leave, then he comes.

Inverse Statement- If he does not come, then we do not leave.

Contrapositive Statement- If we do not leave, then he does not come.

- The given sentence is- “You will qualify GATE only if you work hard.”
- This sentence is of the form- “p only if q”.

So, the symbolic form is **p → q** where-

p : You will qualify GATE

q : You work hard

Converse Statement- If you work hard, then you will qualify GATE.

Inverse Statement- If you will not qualify GATE, then you do not work hard.

Contrapositive Statement- If you do not work hard, then you will not qualify GATE.

- The given sentence is- “If you are intelligent, then you will pass the exam.”
- This sentence is of the form- “If p then q”.

So, the symbolic form is **p → q** where-

p : You are intelligent

q : You will pass the exam

Converse Statement- If you will pass the exam, then you are intelligent.

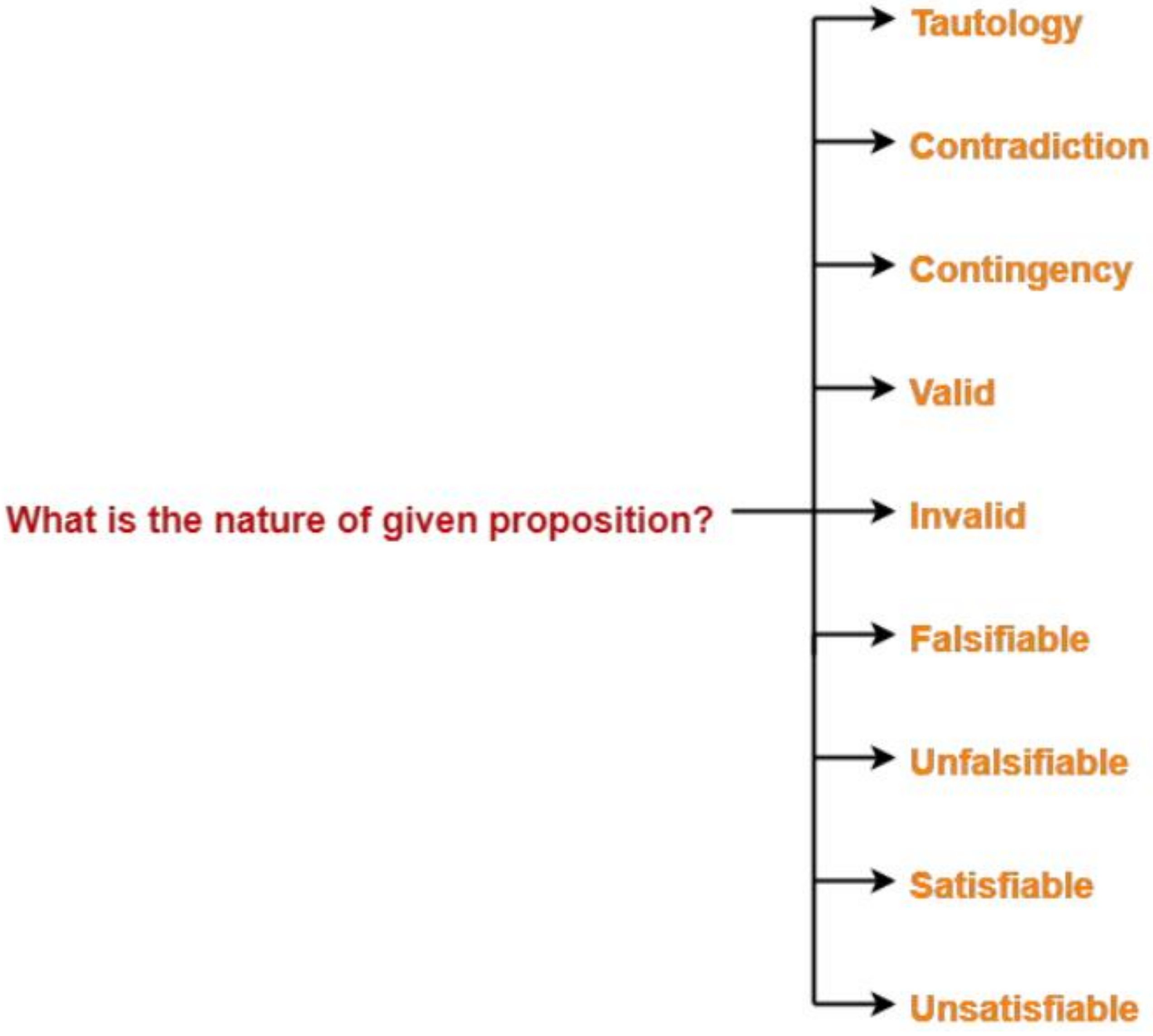
Inverse Statement- If you are not intelligent, then you will not pass the exam.

Contrapositive Statement- If you will not pass the exam, then you are not intelligent.

Determining Nature Of Proposition-

Here,

- We will be given a compound proposition.
- We will be asked to determine the nature of the given proposition.


What is the nature of given proposition?
Tautology
Contradiction
Contingency
Valid
Invalid
Falsifiable
Unfalsifiable
Satisfiable
Unsatisfiable

Tautology-

- A compound proposition is called **tautology** if and only if it is true for all possible truth values of its propositional variables.
- It contains only T (Truth) in last column of its truth table.

Contradiction-

- A compound proposition is called **contradiction** if and only if it is false for all possible truth values of its propositional variables.
- It contains only F (False) in last column of its truth table.

Contingency-

- A compound proposition is called **contingency** if and only if it is neither a tautology nor a contradiction.
- It contains both T (True) and F (False) in last column of its truth table.

Valid-

- A compound proposition is called **valid** if and only if it is a tautology.
- It contains only T (Truth) in last column of its truth table.

Invalid-

- A compound proposition is called **invalid** if and only if it is not a tautology.
- It contains either only F (False) or both T (Truth) and F (False) in last column of its truth table.

Falsifiable-

- A compound proposition is called **falsifiable** if and only if it can be made false for some value of its propositional variables.
- It contains either only F (False) or both T (Truth) and F (False) in last column of its truth table.

Satisfiable-

- A compound proposition is called **satisfiable** if and only if it can be made true for some value of its propositional variables.
- It contains either only T (Truth) or both T (True) and F (False) in last column of its truth table.

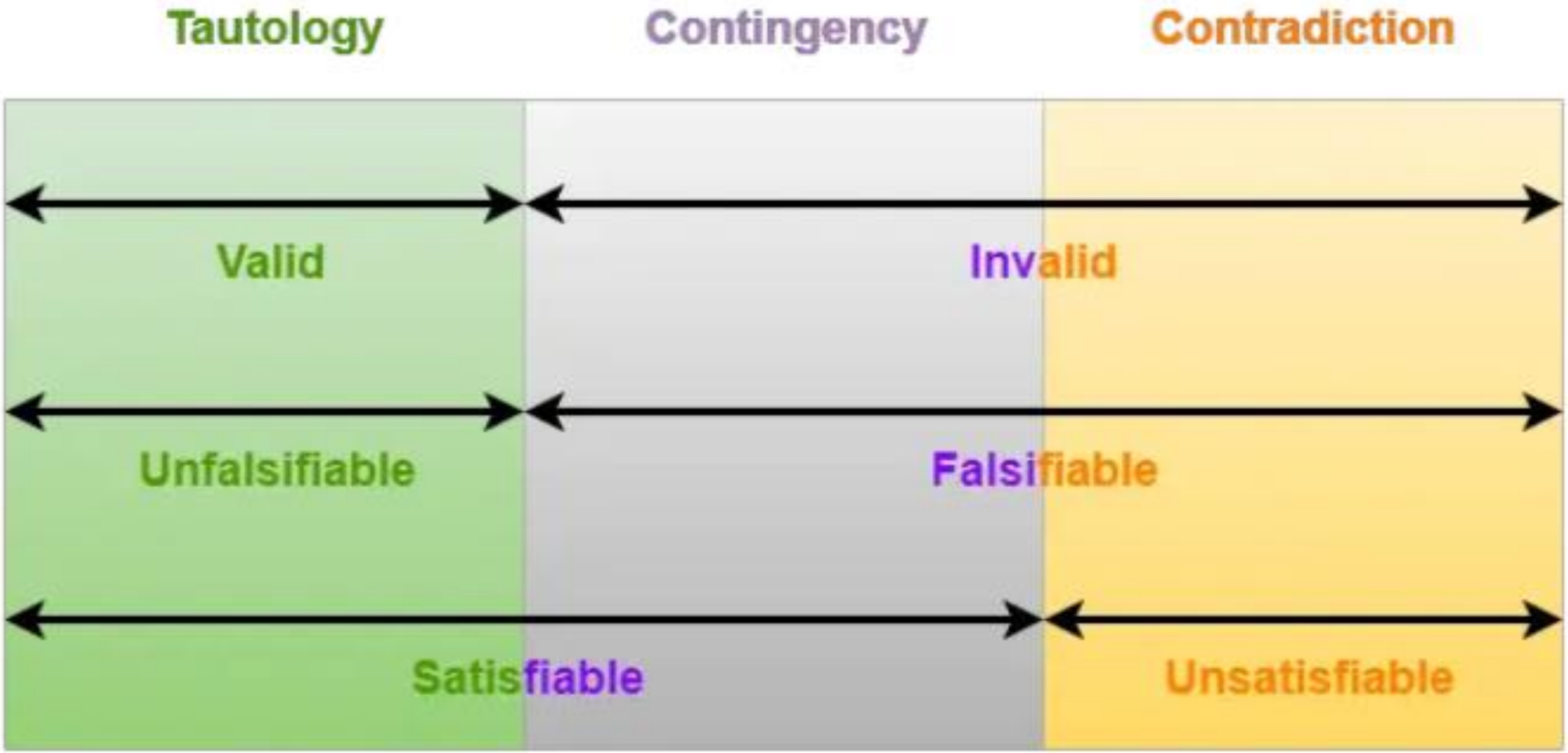
Unsatisfiable-

- A compound proposition is called **unsatisfiable** if and only if it can not be made true for any value of its propositional variables.
- It contains only F (False) in last column of its truth table.

Important Points-

It is important to take a note of the the following points-

- All contradictions are invalid and falsifiable but not vice-versa.
- All contingencies are invalid and falsifiable but not vice-versa.
- All tautologies are valid and unfalsifiable and vice-versa.
- All tautologies are satisfiable but not vice-versa.
- All contingencies are satisfiable but not vice-versa.
- All contradictions are unsatisfiable and vice-versa.


Tautology
Contingency
Contradiction
Valid
Invalid
Unfalsifiable
Falsifiable
Satisfiable
Unsatisfiable

Determine the nature of following propositions-

1. $p \wedge \sim p$
2. $(p \wedge (p \rightarrow q)) \rightarrow \sim q$
3. $[\ (p \rightarrow q) \wedge (q \rightarrow r)\ ] \wedge (\ p \wedge \sim r)$
4. $\sim(p \rightarrow q) \vee (\sim p \vee (p \wedge q))$
5. $(p \leftrightarrow r) \rightarrow (\sim q \rightarrow (p \wedge r))$

Method-01: Using Truth Table-

p	~p	p ∧ ~p
F	T	F
T	F	F

Clearly, last column of the truth table contains only F.

Therefore, given proposition is-

- Contradiction
- Invalid
- Falsifiable
- Unsatisfiable

Method-02: Using Algebra Of Proposition-

- The given proposition is $p \wedge \sim p$
- By complement law, $p \wedge \sim p = F$
- So, given proposition is contradiction, invalid, falsifiable and unsatisfiable.

Method-03: Using Digital Electronics-

In terms of digital electronics,

- The given proposition can be written as p.p’
- Clearly, p.p’ = 0
- So, given proposition is contradiction, invalid, falsifiable and unsatisfiable.

Part-02:

Method-01: Using Truth Table-

p	q	$p \rightarrow q$	$p \wedge (p \rightarrow q)$	$\sim q$	$(p \wedge (p \rightarrow q)) \rightarrow \sim q$
F	F	T	F	T	T
F	T	T	F	F	T
T	F	F	F	T	T
T	T	T	T	F	F

Clearly, last column of the truth table contains both T and F.

Therefore, given proposition is-

- Contingency
- Invalid
- Falsifiable
- Satisfiable

Method-02: Using Algebra Of Proposition-

We have-

$(p \wedge (p \rightarrow q)) \rightarrow \sim q$

$= (p \wedge (\sim p \vee q)) \rightarrow \sim q$ $\{ \because p \rightarrow q = \sim p \vee q \}$

$= \sim(p \wedge (\sim p \vee q)) \vee \sim q$ $\{ \because p \rightarrow q = \sim p \vee q \}$

$= \sim((p \wedge \sim p) \vee (p \wedge q)) \vee \sim q$ { Using Distributive law }

$= \sim(F \vee (p \wedge q)) \vee \sim q$ { Using Complement law }

$= \sim(p \wedge q) \vee \sim q$ { Using Identity law }

$= \sim p \vee \sim q \vee \sim q$ { Using De Morgans law }

$= \sim p \vee \sim q$

- Clearly, the result is neither T nor F.
- So, given proposition is a contingency, invalid, falsifiable and satisfiable.

Method-03: Using Digital Electronics-

We have-

$(p \wedge (p \rightarrow q)) \rightarrow \sim q$

$= (p \wedge (\sim p \vee q)) \rightarrow \sim q$ $\{ \because p \rightarrow q = \sim p \vee q \}$

$= \sim(p \wedge (\sim p \vee q)) \vee \sim q$ $\{ \because p \rightarrow q = \sim p \vee q \}$

Now in terms of digital electronics, we have-

$= (p.(p' + q))' + q'$

$= (p.p' + p.q)' + q'$

$= (p.q)' + q'$ $\{ \because p.p' = 0 \}$

$= p' + q' + q'$ { Using De Morgan's law }

$= p' + q'$

- Clearly, the result is neither 0 nor 1.
- So, given proposition is a contingency, invalid, falsifiable and satisfiable.

Method-01: Using Truth Table-

Let [$(p \to q) \wedge (q \to r)$] $\wedge$ ($p \wedge \sim r$) = R (say)

p	q	r	$p \to q$	$q \to r$	$(p \to q) \wedge (q \to r)$	$p \wedge \sim r$	R
F	F	F	T	T	T	F	F
F	F	T	T	T	T	F	F
F	T	F	T	F	F	F	F
F	T	T	T	T	T	F	F
T	F	F	F	T	F	T	F
T	F	T	F	T	F	F	F
T	T	F	T	F	F	T	F
T	T	T	T	T	T	F	F

Clearly, last column of the truth table contains only F.

Therefore, given proposition is-

- Contradiction
- Invalid
- Falsifiable
- Unsatisfiable

Predicate Logic

- To overcome these limitations, predicate logic introduces:
  - **Variables** for dynamic queries.
  - **Quantifiers** (`forall`, `exists`) for general rules.
  - **Advanced Reasoning** for more intelligent systems.

Predicate Calculus

- **Propositional Calculus:** Uses atomic symbols (e.g., P, Q) to represent entire propositions. Lacks the ability to describe components of propositions.
- **Predicate Calculus:** Allows representation of relationships between objects and properties, using predicates and variables.

Features

- **Predicates:**
  - Represents a relationship between objects.
  - Example: `weather(tuesday, rain)`.
- **Variables:**
  - Generalize assertions about classes of objects.
  - Example: `∀X (weather (X, rain))` → "It rains every day."
- **Inference Rules:**
  - Access and manipulate predicate calculus expressions.

Syntax of Predicate Calculus

- **Symbols:**
  - Alphabet: Letters (`A-Z, a-z`), Digits (`0-9`), Underscore `(_).`
  - Symbols start with a letter and may contain legal characters.
- **Types of Symbols:**
  - **Constants:** Specific objects, properties (lowercase). Example: george, blue.
  - **Variables:** Represent classes of objects (uppercase). Example: X, Day.
  - **Functions:** Map elements from a domain to a range (lowercase). Example: father(david).
  - **Predicates:** Define relationships. Example: likes(george, kate).
- **Reserved Truth Symbols:**
  - true, false.

Well-Formed Expressions

- **Constants:** Start with a lowercase letter.
    - Example: `blue, rain.`
- **Variables:** Start with an uppercase letter.
    - Example: `X, Day.`
- **Function Expressions:** Function symbol followed by arguments.
    - Example: `father(david), price(bananas).`
- **Atomic Sentences:** Predicate followed by arguments (arity matters).
    - Example: `likes(george, kate), friends(bill, george).`

Evaluation

- Replacing a function with its value is called **evaluation**.
  - **Example**: If `father(david) = george`, then `friends(father(david), allen)` evaluates to `friends(george, allen)`.

Definitions

- **Terms:** Constants, variables, or function expressions.
  - Example: `X, mother(sarah), cat.`
- **Predicates:** Define relations of arity n.
  - Example: `likes(X, Y), friends(bill, george).`

Examples

- **Predicate with constants:** `likes(george, kate).`
- **Predicate with variables:** `friends(X, Y).`
- **Functions as arguments:** `friends(father(david), father(andrew)).`

Core Concepts

- **Predicate Symbols and Atomic Sentences**:
  - Predicate symbols begin with lowercase letters.
  - Predicates have an **arity**, which defines the number of arguments (e.g., `mother(eve, abel)` has arity 2).
  - An **atomic sentence** is formed by applying a predicate to the correct number of terms enclosed in parentheses and separated by commas.
  - **Truth values** `(true, false)` are also considered atomic sentences.
- **Logical Connectives**:
  - The connectives `∧ (and), ∨ (or), ¬ (not), → (implies), and ≡ (equivalent)` are used to combine atomic sentences into more complex expressions.

Core Concepts

- **Quantifiers**:
  - **Universal Quantifier** (∀): Indicates that a sentence applies to all elements in the domain.
    - Example: `∀ X likes(X, ice_cream)` means "everyone likes ice cream."
  - **Existential Quantifier** (∃): Indicates that a sentence applies to at least one element in the domain.
    - Example: `∃ Y friends(Y, peter)` means "someone is a friend of Peter."
- **Defining Complex Sentences**: Sentences in predicate calculus can be built recursively:
  - Base: Atomic sentences.
  - Negation: If s is a sentence, so is ¬s.
  - Conjunction/Disjunction: If `s1` and `s2` are sentences, so are `s1 ∧ s2` and `s1 ∨ s2`.
  - Implication/Equivalence: If `s1` and `s2` are sentences, so are `s1 → s2` and `s1 ≡ s2`.
  - Quantification: If X is a variable and s a sentence, ∀X s and ∃X s are also sentences.

Example: Biblical Genealogy

Using predicates to describe relationships:

- **Atomic predicates:**
  - `mother(eve, abel)`
  - `mother(eve, cain)`
  - `father(adam, abel)`
  - `father(adam, cain)`

- **Derived relationships:**
  - Parent definition: $\forall X \forall Y\ (\mathbf{father}(X, Y) \lor \mathbf{mother}(X, Y) \rightarrow \mathbf{parent}(X, Y))$
  - Sibling definition: $\forall X \forall Y \forall Z\ (\mathbf{parent}(X, Y) \land \mathbf{parent}(X, Z) \rightarrow \mathbf{sibling}(Y, Z))$
  - Inference: From these rules, we can deduce `sibling(cain, abel)`.

Knowledge Representation in AI

- Knowledge Representation (KR) is a key aspect of AI that deals with **how information is structured, stored, and used for reasoning**.
- It defines how machines understand and manipulate knowledge to **make decisions, infer conclusions, and interact with their environment**.

Types of Knowledge in AI

AI systems need various types of knowledge to function effectively. These include:

- **Declarative Knowledge (Facts and Descriptions)**
  - Represents explicit knowledge about objects, concepts, and facts.
  - Example: "A dog is a mammal."
- **Procedural Knowledge (Rules and Procedures)**
  - Describes **how to do something** rather than just stating facts.
  - Example: "To drive a car, press the accelerator to move forward."

Types of Knowledge in AI

- **Metaknowledge (Knowledge About Knowledge)**
  - Helps AI decide **which knowledge is relevant** in a given situation.
  - Example: "Experts in medicine rely on research studies more than opinions."
- **Heuristic Knowledge (Experience-Based Rules of Thumb)**
  - Uses practical approaches for problem-solving.
  - Example: "If traffic is heavy on Main Street, take an alternate route."
- **Structural Knowledge (Relationships and Connections)**
  - Represents **how different concepts relate** to each other.
  - Example: "A square is a type of rectangle, and both are polygons."

Techniques for Knowledge Representation

There are four primary techniques used in AI to represent knowledge:

- **Logical Representation (Propositional and Predicate Logic)**
  - Uses **formal logic** to express relationships and rules.
  - Example: "If it is raining, then the ground is wet."
- **Semantic Networks (Graph-Based Representation)**
  - Represents knowledge as **a network of nodes and edges**.
  - Example: A "bird" is connected to "has wings" and "can fly."

Techniques for Knowledge Representation

- **Frames (Structured Knowledge Representation)**
  - Represents knowledge in **hierarchical frames** with slots and values.
  - Example: A **car** has slots for "brand," "color," "fuel type," etc.
- **Production Rules (IF-THEN Rules)**
  - Uses condition-action pairs for decision-making.
  - Example: "If a patient has a high fever, then suggest a blood test."

Issues in Knowledge Representation

- **Expressiveness vs. Efficiency**
  - Highly expressive representations can be complex and slow.
  - Example: Propositional logic is simple but lacks flexibility.
- **Handling Uncertainty**
  - Many real-world problems involve **incomplete or uncertain knowledge**.
  - Solution: **Probabilistic reasoning or fuzzy logic.**
- **Scalability**
  - As knowledge grows, maintaining **consistency and efficiency** becomes difficult.
- **Common Sense Knowledge**
  - AI lacks the ability to make **obvious inferences** that humans can.
  - Example: If a book falls, it will hit the ground (AI might need explicit rules).

Forward and Backward Chaining

- Inference mechanisms like **forward chaining and backward chaining** are fundamental in AI, especially in **rule-based systems** and **expert systems**. These techniques help AI derive conclusions from a knowledge base.

Forward Chaining (Data-Driven Inference)

- Forward chaining is a **bottom-up reasoning process** that starts with known facts and **applies rules** to infer new facts.
- **How It Works**
  1. **Start with a set of known facts**.
  2. **Check which rules can be applied** (IF condition matches known facts).
  3. **Apply the rule and derive new facts**.
  4. **Repeat until no more new facts can be inferred** or until a goal is reached.

Example: Diagnosing a Disease

- We have an **expert system for medical diagnosis** that follows this rule set:
- **Knowledge Base (Rules)**
  - **Rule 1**: IF a patient has a fever AND cough, THEN the patient might have the flu.
  - **Rule 2**: IF a patient has a flu, THEN suggest taking rest and fluids.
  - **Rule 3**: IF a patient has a flu AND breathing issues, THEN recommend seeing a doctor.
- **Known Facts (Initial Information)**
  - The patient has a fever.
  - The patient has a cough.

Example: Diagnosing a Disease

- **Step-by-Step Process**
- **Applying Rule 1**
  - Fact: Patient has fever AND cough.
  - Conclusion: The patient might have the flu (new fact inferred).
- **Applying Rule 2**
  - New Fact: Patient has the flu.
  - Conclusion: Suggest taking rest and fluids.
- **Applying Rule 3**
  - No breathing issues, so we **do not** apply this rule.
- **Final Outcome:** The system infers that the patient has the flu and recommends rest and fluids.

Backward Chaining (Goal-Driven Inference)

- Backward chaining is a **top-down reasoning process** that starts with a **goal** and works **backward** to check if supporting facts exist.
- **How It Works**
  1. **Start with a goal or hypothesis**.
  2. **Look for rules that support this goal**.
  3. **Check if the conditions of these rules are met** using existing facts.
  4. **If conditions are not met, keep working backward** until known facts confirm or reject the goal.

Example: AI in Medical Diagnosis (Backward Chaining Approach)

- Imagine a **medical diagnosis system** that helps doctors determine whether a patient has **pneumonia**.
- **Step 1: Define the Goal (Hypothesis to Verify)**
  - **Does the patient have pneumonia?**
- **Step 2: Look for Rules That Can Prove This Goal**
  - The system checks its knowledge base for **rules related to pneumonia**.
  - It finds this rule:
  - **Rule 1:** If a patient has **chest pain, fever, and difficulty breathing**, then they might have **pneumonia**.

Example: AI in Medical Diagnosis (Backward Chaining Approach)

- **Step 3: Work Backward to Find Supporting Facts**
  - **Does the patient have chest pain?**
  - The AI asks the patient or checks medical records.
  - Answer: **Yes**
- **Does the patient have a fever?**
  - AI checks recent temperature readings.
  - Answer: **Yes**
- **Does the patient have difficulty breathing?**
  - AI analyzes **past diagnoses, symptoms, or a breathing test**.
  - Answer: **Yes**
- **Step 4: Since All Conditions Are Met, AI Confirms the Diagnosis**
  - **Conclusion:** The patient is diagnosed with **pneumonia**.

Example: Finding a Suspect in an Investigation

- Imagine a detective trying to identify the criminal in a case.
- **Goal:**
  - Find out if **John** is the thief.
- **Knowledge Base (Rules)**
  - **Rule 1**: IF a person was seen at the crime scene, THEN they are a suspect.
  - **Rule 2**: IF a suspect was seen stealing, THEN they are the thief.
  - **Rule 3**: IF a suspect has stolen goods, THEN they are the thief.

Example: Finding a Suspect in an Investigation

- **Step-by-Step Process**
- **Start with the goal:** Is John the thief?
- **Check Rule 2**:
  - Was John seen stealing? **No evidence.**
  - Move backward to Rule 1.
- **Check Rule 1**:
  - Was John seen at the crime scene? **Yes (fact).**
  - Now, John is a **suspect** (new fact).
- **Check Rule 3**:
  - Does John have stolen goods? **Yes (fact).**
  - **Conclusion:** John is the thief.
- Thus, the backward chaining process confirms **John is guilty** by **working backward from the goal** and verifying the supporting evidence.

Comparison: Forward vs. Backward Chaining

Feature	Forward Chaining	Backward Chaining
Starts With	Known facts	A goal (hypothesis)
Process	Applies rules to infer new facts	Traces back to find supporting facts
Best Used For	Large datasets, automated decision-making (e.g., expert systems, recommendation systems)	Problem-solving, diagnosis, planning (e.g., medical diagnosis, detective work)
Example	Inferring flu from symptoms	Checking if John is a thief

Example: AI in Fraud Detection (Forward and Backward Chaining)

- **Using Forward Chaining**, we find **what type of fraud might be happening**.
- **Using Backward Chaining**, we verify **if a transaction is fraudulent** by working backward from the goal.

Forward Chaining: Identifying the Type of Fraud

- **Knowledge Base (Rules)**
  - **Rule 1**: If a transaction **exceeds $10,000** and is **from an unknown device**, THEN it **might be money laundering**.
  - **Rule 2**: If a transaction **occurs from two countries within 5 minutes**, THEN it **might be account takeover fraud**.
  - **Rule 3**: If a transaction **comes from a blocked IP address**, THEN it **might be a phishing attack**.

Forward Chaining: Identifying the Type of Fraud

- **Known Facts (Initial Data)**
  - The transaction **exceeded $10,000**.
  - The transaction **came from an unknown device**.
- **Step-by-Step Process**
  - **Applying Rule 1**:
    - Fact: Transaction **exceeded $10,000** and **came from an unknown device**.
    - Conclusion: **The transaction might be money laundering**.
  - **Other rules are checked, but no other conditions match.**

Backward Chaining: Confirming If a Transaction Is Fraudulent

- **Goal (Hypothesis to Verify)**
  - **Is this transaction fraudulent?**
- **Knowledge Base (Rules)**
  - **Rule 1**: If a transaction is **money laundering**, THEN it is **fraudulent**.
  - **Rule 2**: If a transaction is **account takeover fraud**, THEN it is **fraudulent**.
  - **Rule 3**: If a transaction **matches known fraud patterns**, THEN it is **fraudulent**.

Backward Chaining: Confirming If a Transaction Is Fraudulent

- **Step-by-Step Process**
- **Start with the goal:**
  - Is this transaction fraudulent?
- **Check Rule 1:**
  - Does it match a known fraud type?
  - AI checks if this transaction was flagged as **money laundering** in the forward chaining process.
  - **Yes** → It is fraudulent.
- **No need to check other rules, since fraud is confirmed.**