Knowledge Based Agent

Chapter – 04

Knowledge Representation and Logic

4 Knowledge and Reasoning

4.1 Knowledge based Agents, Brief Overview of propositional logic, First Order Logic: Syntax and Semantic, Inference in FOL, Forward chaining, backward Chaining.

4.2 Knowledge Engineering in First-Order Logic, Unification, Resolution

4.3 Uncertain Knowledge and Reasoning: Uncertainty, Representing knowledge in an uncertain domain, The semantics of belief network, Simple Inference in belief network

Content

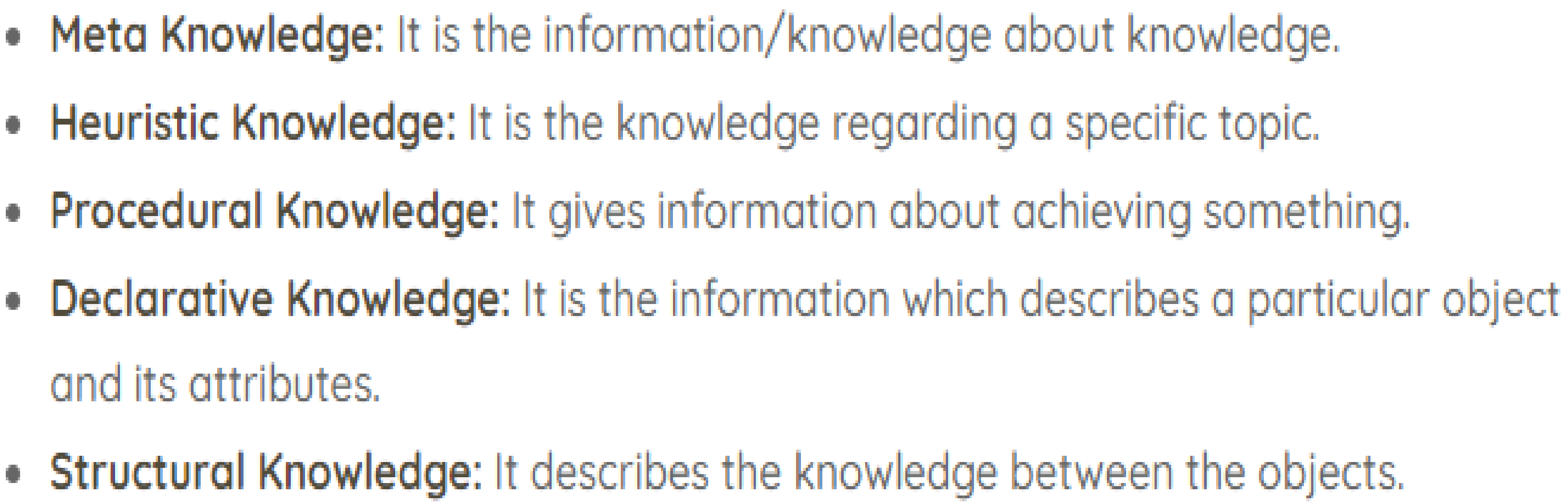
- What is knowledge ? partially observable environment

- Importance of Knowledge representation in Intelligent agents

- Use of propositional logic, predicate logic, First order logic – syntax, semantics, validity , entailment etc

- Represent natural language statements in logic

- Deduct new sentences by applying inference rules.



- **Meta Knowledge:** It is the information/knowledge about knowledge.
- **Heuristic Knowledge:** It is the knowledge regarding a specific topic.
- **Procedural Knowledge:** It gives information about achieving something.
- **Declarative Knowledge:** It is the information which describes a particular object and its attributes.
- **Structural Knowledge:** It describes the knowledge between the objects.

What is Knowledge?

- Knowledge is **what I know** and Information is **what we know**

- Knowledge can be considered as the distillation of information that has been **collected, classified, organized, integrated, abstracted and value added**.

- Intelligent behavior is not dependent so much on the methods of reasoning as on the knowledge one has to reason with.

Characteristics of Knowledge

- –Knowledge is **huge** (Large in number or quantity).
- –Knowledge is **hard to characterize** accurately.
- –Knowledge differs from data in that it is organized such that it corresponds to the ways it will be used.
- –Knowledge is **interpreted differently** by **different people.**

Knowledge

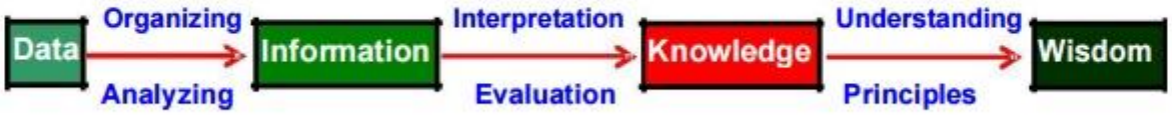


Fig 1 Knowledge Progression

- **Data** : Collection of **disconnected facts**
- **Information** :Emerges when **Relationships between facts** are established and understood. Provides answer to **Who, what, where and when.**
- **Knowledge** : emerges when **relationship between patterns** are identified and understood. Provides answer to **"How"**
- **Wisdom** : is the pinnacle of understanding, uncovers the **principles of relationships that describe patterns.** Provides answer as **" Why"**

Knowledge Model

- **Data and information** : Past data
- **Knowledge :** Present/ enable us to perform
- **Wisdom :** Future – acquire vision for what will be

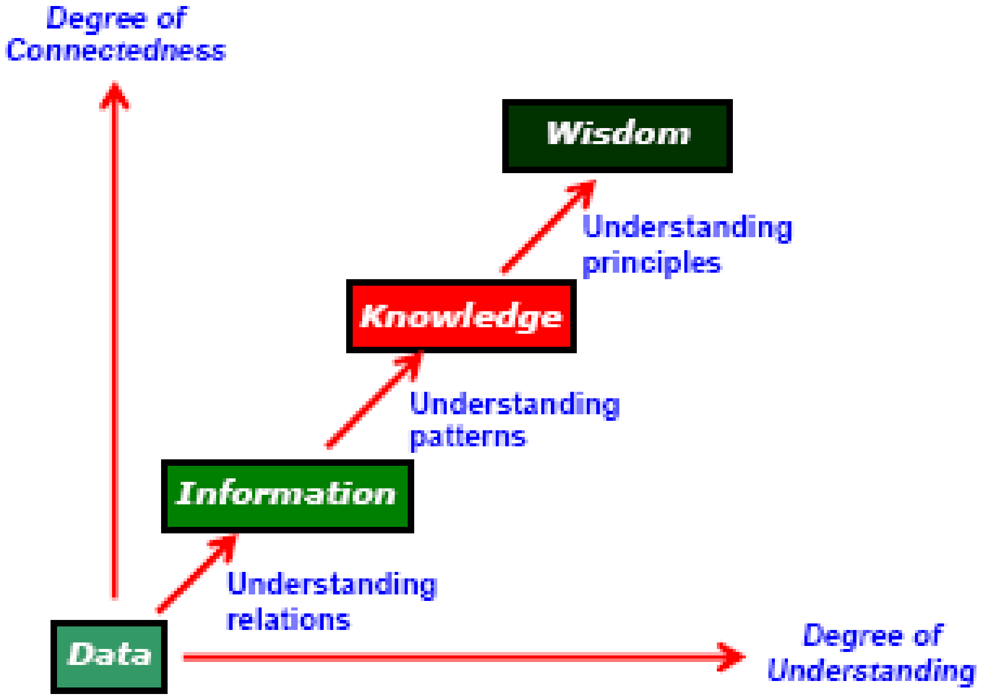
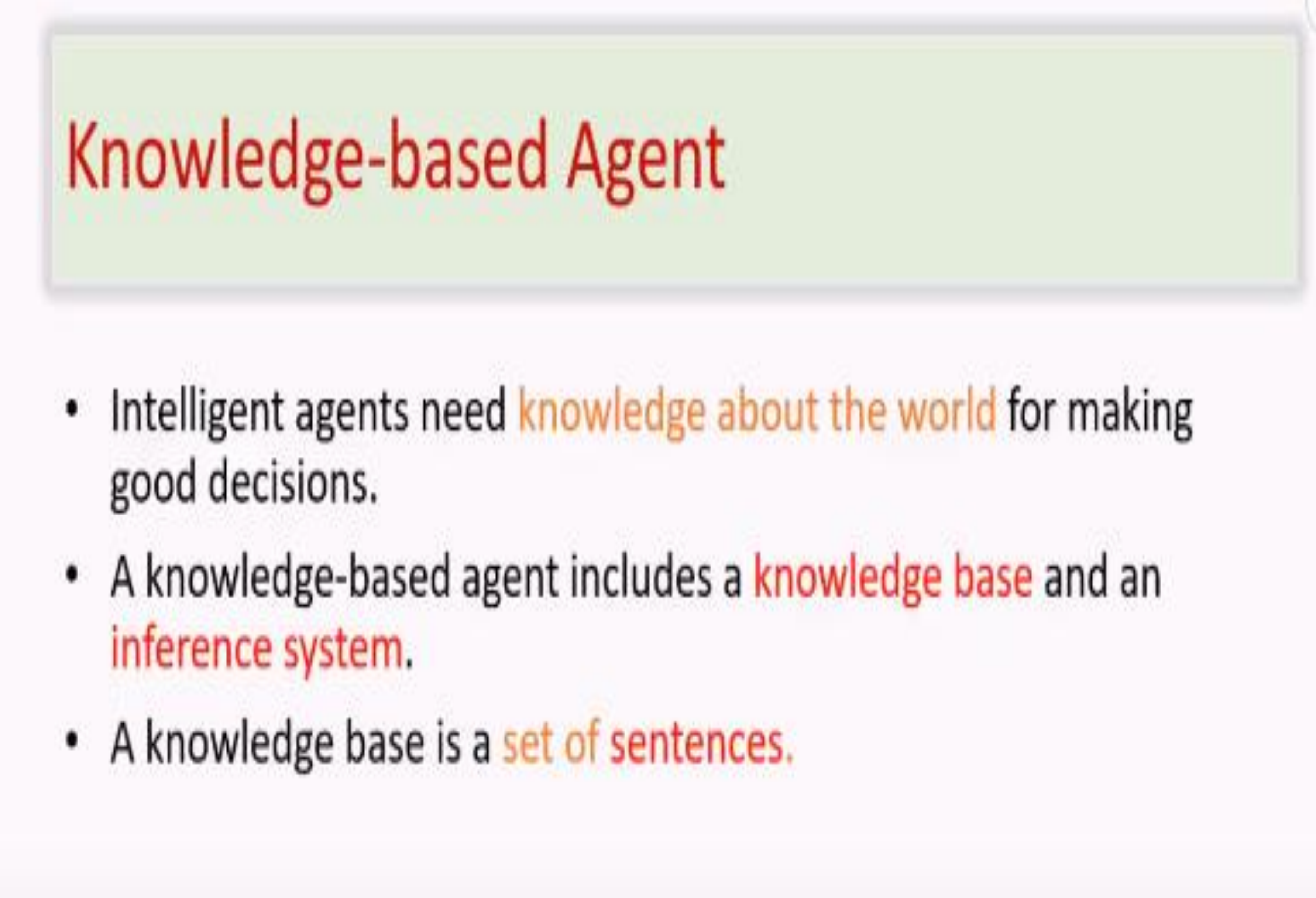


Fig. Knowledge Model

Knowledge based agents / Intelligent agents

Importance of Knowledge representation in Intelligent Agents

- **Intelligent agents** should have capacity for
  - **Perceiving** : acquiring information from environment
  - **Knowledge Representation** : representing its understanding of the world
  - **Reasoning :** inferring the implications of what it knows and of the choices it has
  - **Acting** : choosing what it want to do and carry it out.



Knowledge-based Agent

- Intelligent agents need knowledge about the world for making good decisions.
- A knowledge-based agent includes a knowledge base and an inference system.
- A knowledge base is a set of sentences.

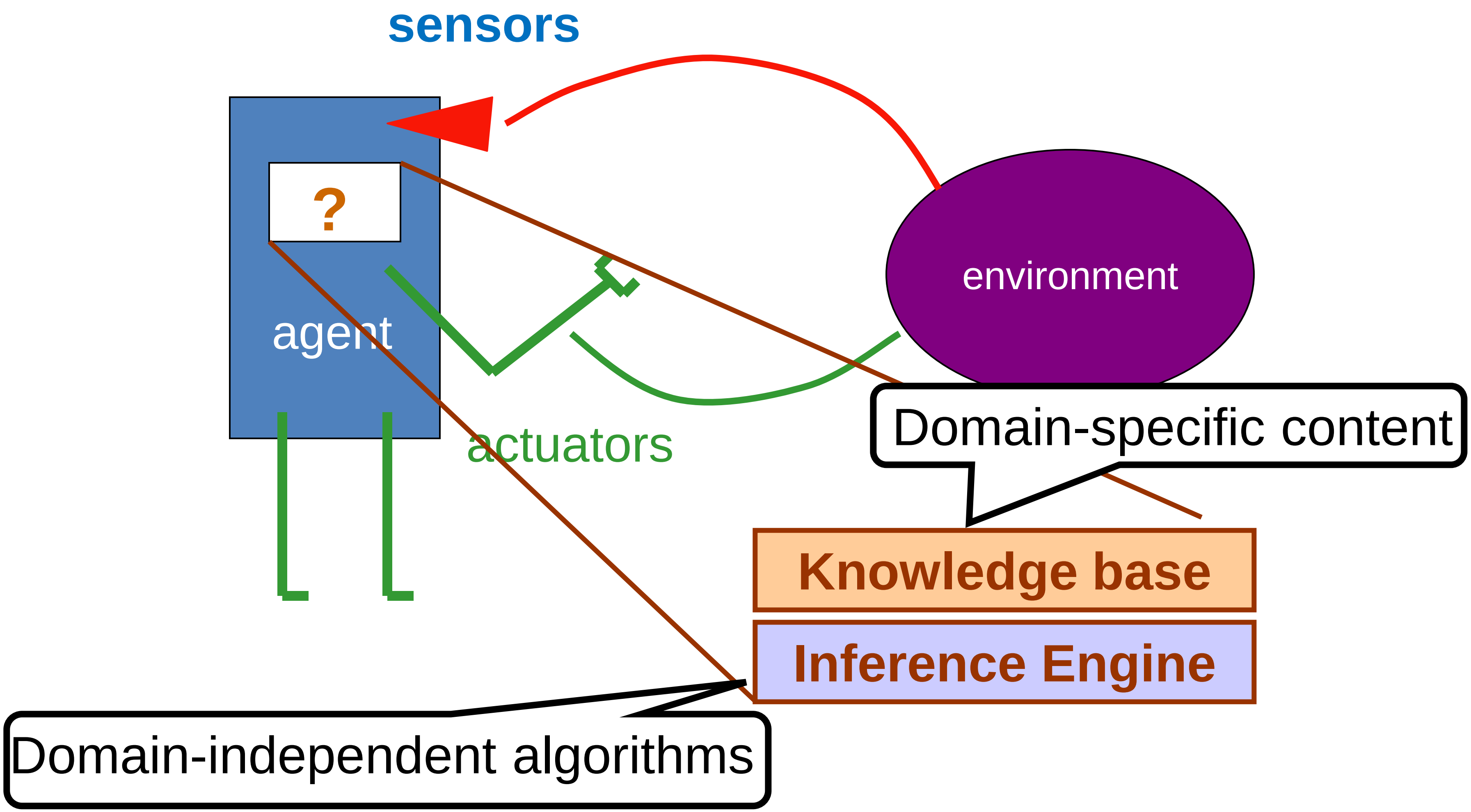
What Agents do ?

- It TELLs the knowledge base what it perceives.
- It ASKs the knowledge base what action it should perform.
- It performs the chosen action.

Types of Knowledge

- **Implicit type : exist within human being**
  - **Tacit / unconscious knowledge –** cannot be expressed in language . Exist with human beings, cannot copied, drawn from experiences, subjective insight Skills
- **Explicit Type : exist outside human being, can be shared ,copied, stored etc**
  - **Procedural knowledge** - knowing how to do something , e.g. Ram is older or John ? Soln – find their ages ? - algorithm
  - **Declarative knowledge**– Statement that can be True/false – Specification

Knowledge Base Agent



sensors

?

agent

actuators

environment

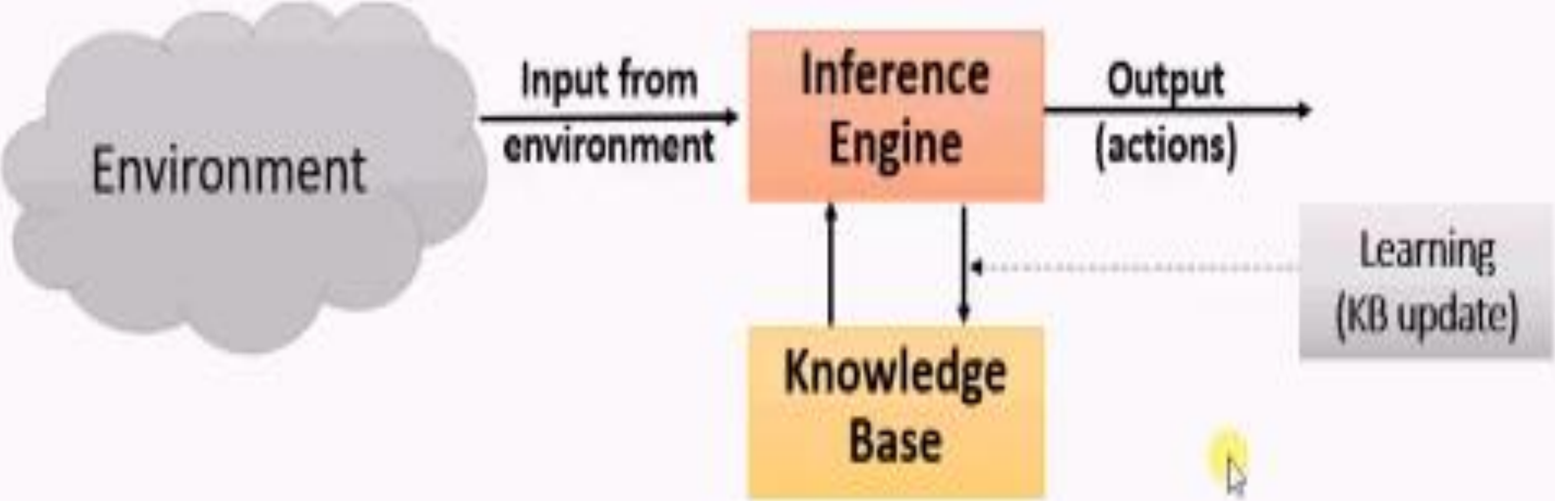
Domain-specific content

Knowledge base

Inference Engine

Domain-independent algorithms

Architecture of KBA



Environment

Input from environment

Inference Engine

Output (actions)

Knowledge Base

Learning (KB update)

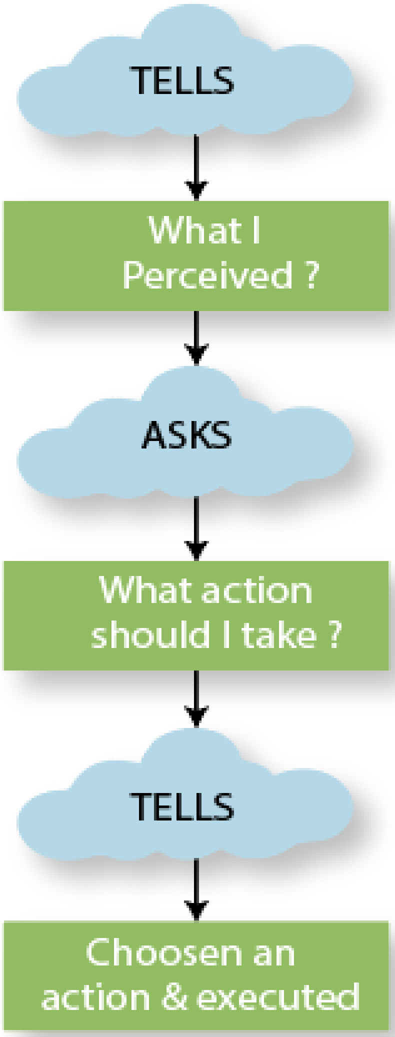
Logical Agents

- **Logic (Knowledge-Based) agents** combine **general knowledge with current percepts** to infer hidden aspects of current state prior to selecting actions
  - **Crucial in partially observable environments**

Knowledge Base Agent

- Inference - TELL, ASK
- Knowledge base

Each time the agent program is called, it does three things. **First, it TELLs the knowledge base what it perceives.** **Second, it ASKs the knowledge base what action it should perform**. *In the process of answering this query, extensive reasoning may be done about the current state of the world, about the outcomes of possible action sequences, and so on.* **Third, the agent program TELLs the knowledge base which action was chosen, and the agent executes the action.**



TELLS

What I Perceived ?

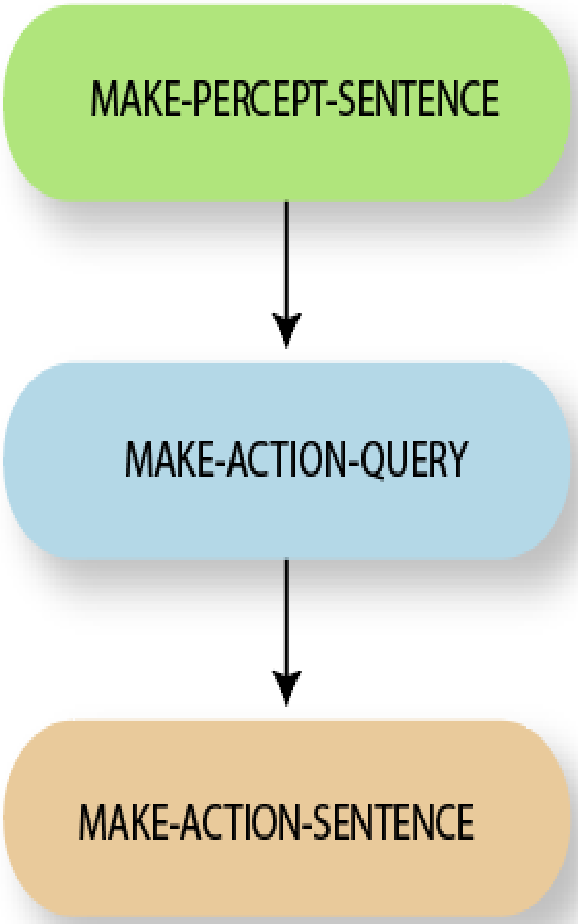
ASKS

What action should I take ?

TELLS

Choosen an action & executed

Mechanism of an Agent Program



MAKE-PERCEPT-SENTENCE

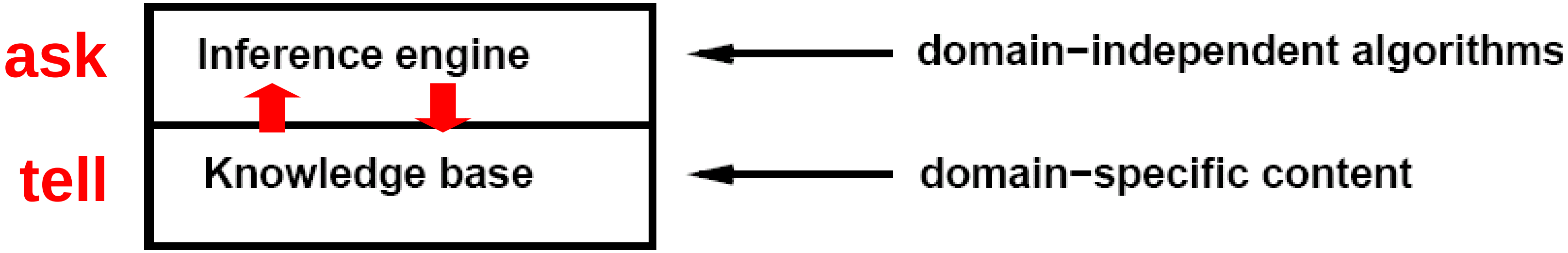
MAKE-ACTION-QUERY

MAKE-ACTION-SENTENCE

Functions hiding details of Knowledge Representation Language

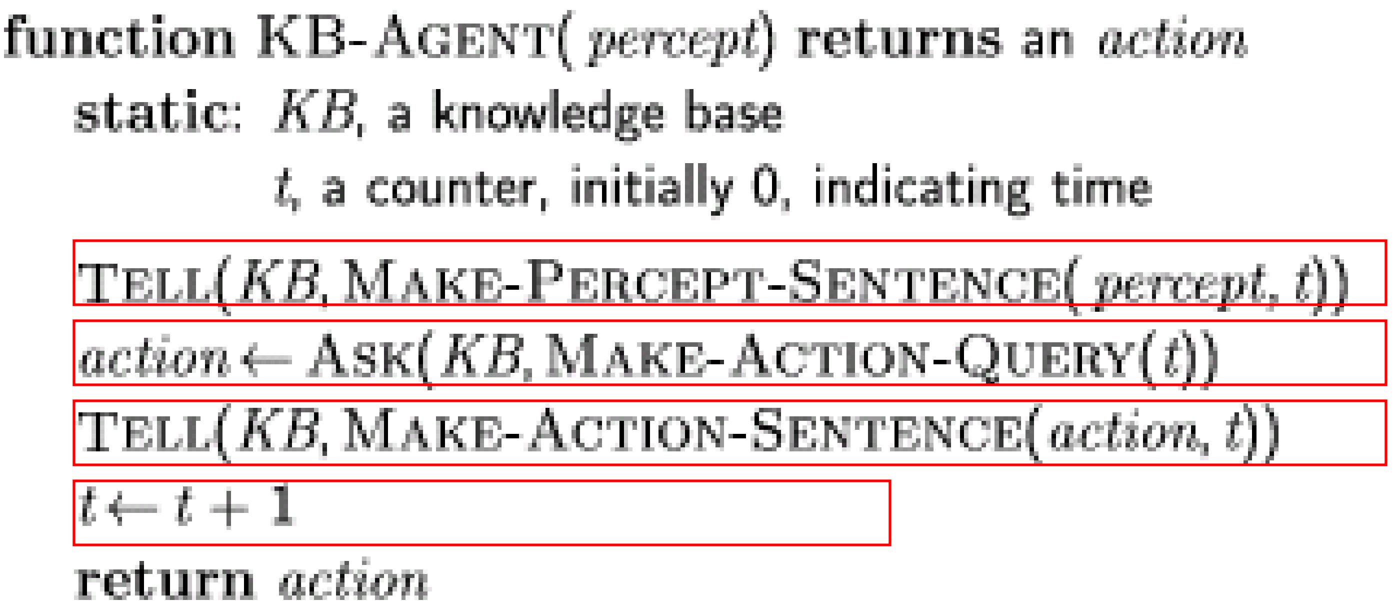
Knowledge Base

- **Knowledge Base:** set of sentences represented in a knowledge representation language and represents assertions about the world.



- **Inference rule:** when **one ASKs questions** of the KB, the answer should ***follow*** from what has been **TELLed** to the **KB previously**.

Generic Knowledge Base (KB) Based Agent

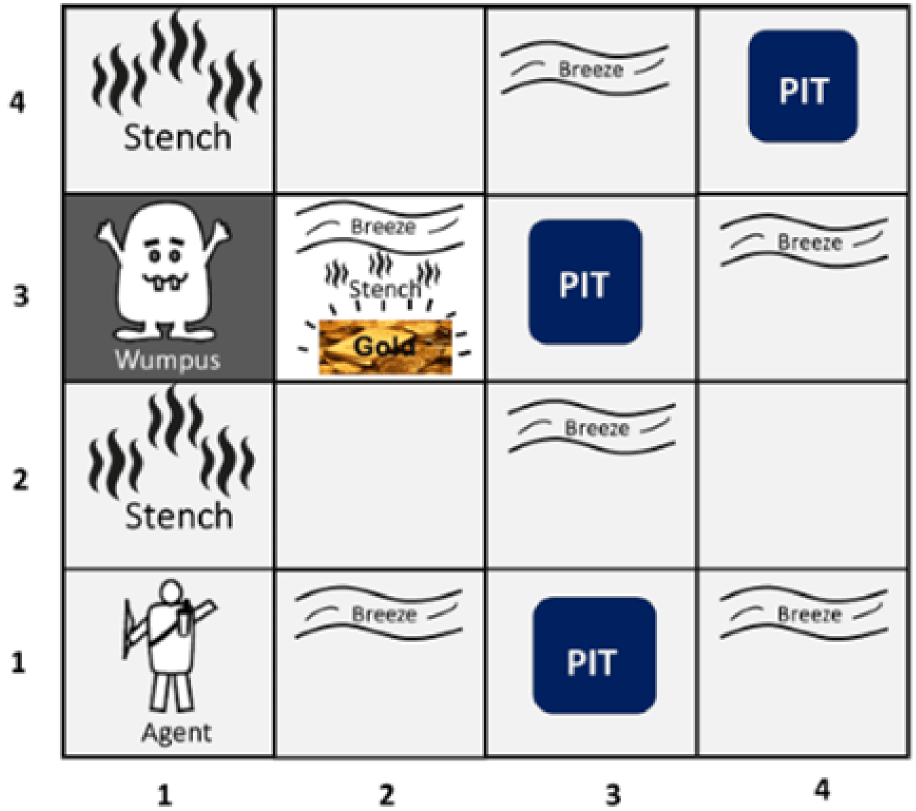


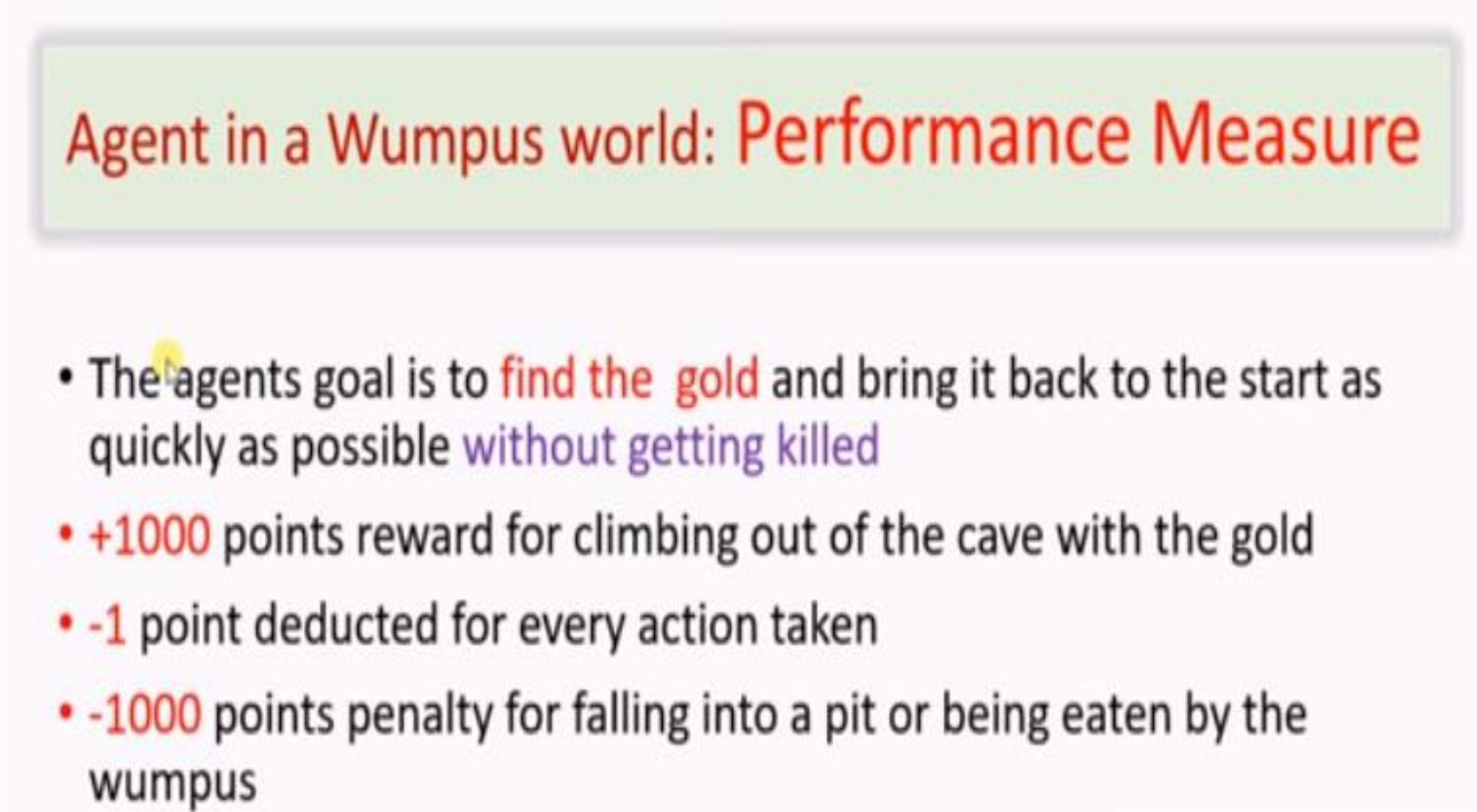
```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time
    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action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

1. It **TELL’s KB** what it perceives
2. It **ASK’s** the KB what action it should perform –
3. Once the choice is made, agent **records its choice with TELL** and **executes** the action.

WUMPUS WORLD PROBLEM

KB Agent

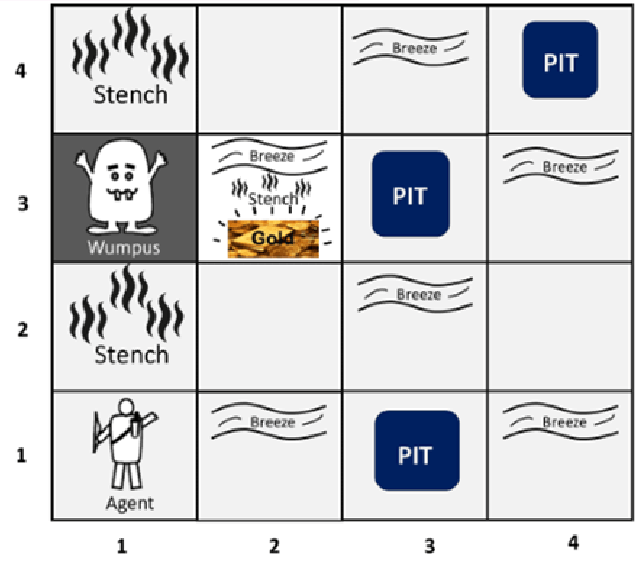




Agent in a Wumpus world: Performance Measure

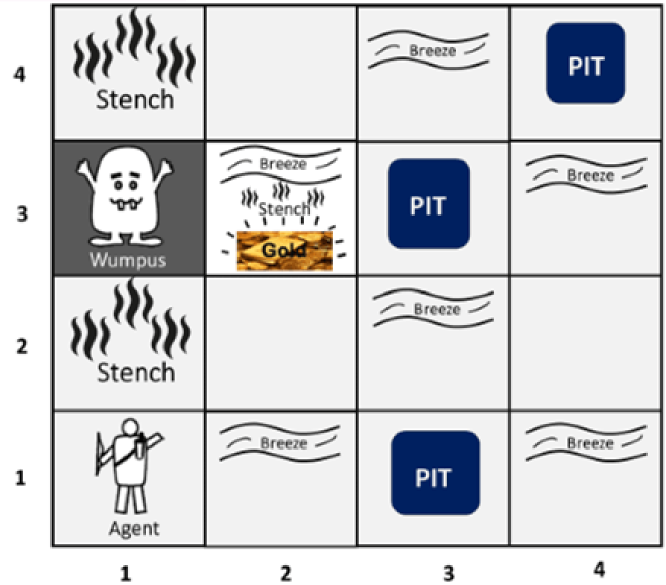
- The agents goal is to find the gold and bring it back to the start as quickly as possible without getting killed
- +1000 points reward for climbing out of the cave with the gold
- -1 point deducted for every action taken
- -1000 points penalty for falling into a pit or being eaten by the wumpus

For using arrow: -10



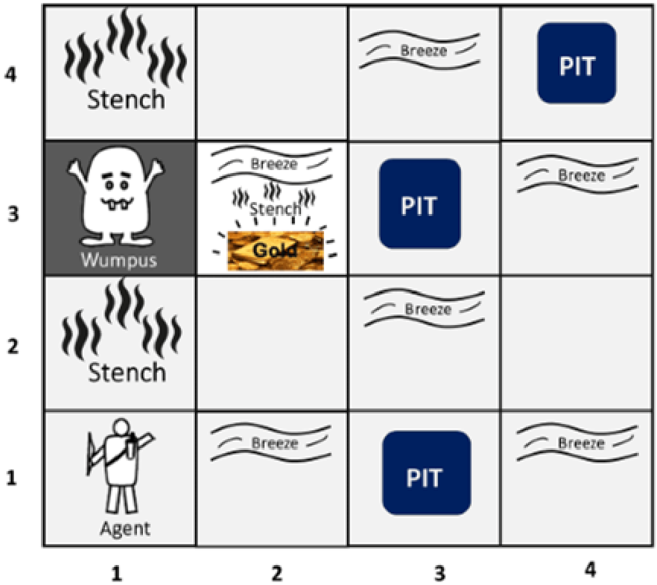
Agent in a Wumpus world: Environment

- A 4×4 grid of rooms
- The agent always starts in the square [1,1], facing to the right
- The locations of the gold and the wumpus are chosen randomly
- In addition, each square other than the start can be a pit, with probability 0.2



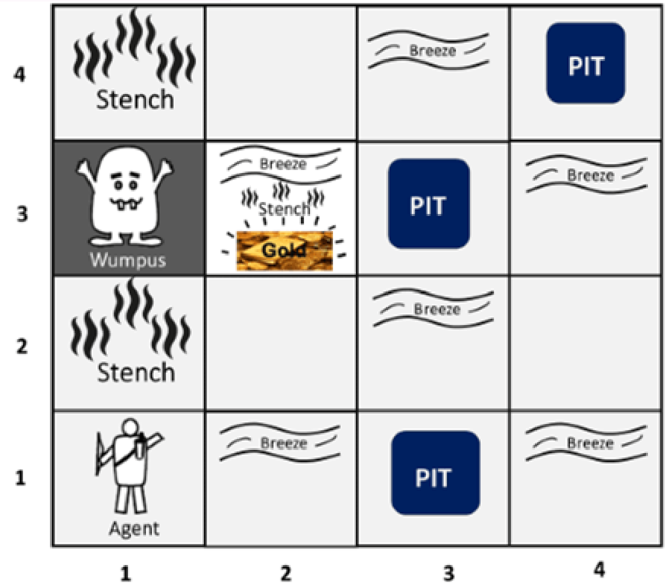
Wumpus PEAS description

- Sensors: Stench, Breeze, Glitter, Bump, Scream
- Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot



Agent in a Wumpus world: Actuators

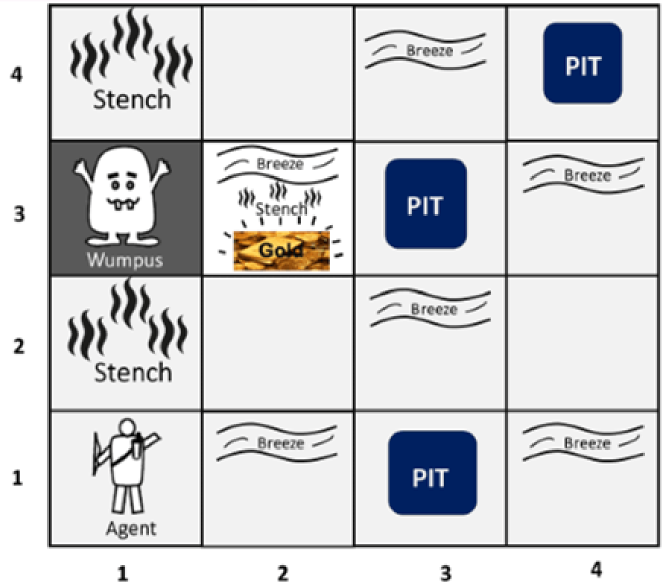
- **go forward**
- **turn right** 90 degrees
- **turn left** 90 degrees
- **grab** means pick up an object that is in the same square as the agent
- **shoot** means fire an arrow in a straight line in the direction the agent is looking.
  - The arrow continues until it either hits and kills the wumpus or hits the wall.
  - The agent has only one arrow.
  - Only the first shot has any effect.
- **climb** is used to leave the cave.
  - Only effective in start field.
- **die**, if the agent enters a square with a pit or a live wumpus.
  - (No take-backs!)



Agent in a Wumpus world: Sensors

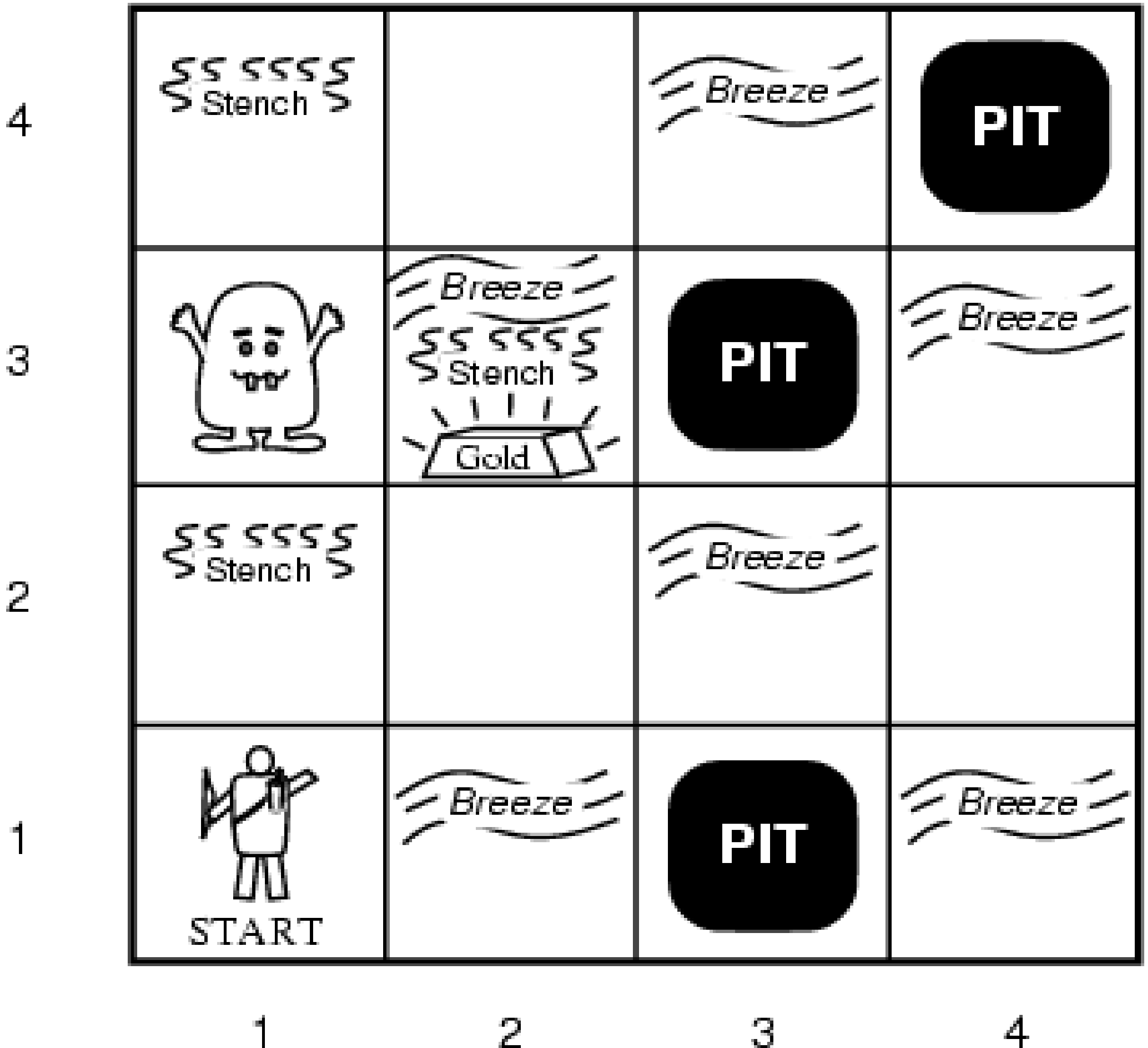
- The agent perceives
  - A stench in the square containing the wumpus and in the adjacent squares (not diagonally).
  - A breeze in the squares adjacent to a pit
  - A glitter in the square where the gold is
  - A bump, if it walks into a wall
  - A scream everywhere in the cave, if the wumpus is killed
- The percepts will be given as a five-symbol list:
  - If there is a stench, and a breeze, but no glitter, no bump, and no scream, the percept is

    [Stench, Breeze, None, None, None]
- The agent can not perceive its own location.



A typical Wumpus world

- The agent always **starts in [1,1].**
- The task of the agent is **to find the gold, return to the field [1,1] and climb out of the cave.**



1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

A	= *Agent*
B	= *Breeze*
G	= *Glitter, Gold*
OK	= *Safe square*
P	= *Pit*
S	= *Stench*
V	= *Visited*
W	= *Wumpus*

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 A B OK	3,1 P?	4,1

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Wumpus World Characterization

- **Observable?**
  - No, only local perception
- **Deterministic?**
  - Yes, outcome exactly specified
- **Static?**
  - Yes, Wumpus and pits do not move
- **Discrete?**
  - Yes
- **Single-agent?**
  - Yes, Wumpus is essentially a natural feature.

Rules and Atomic Propositions

- Some Atomic Propositions

  S12 = There is a stench in cell (1,2)

  B34 = There is a breeze in cell (3,4)

  W22 = Wumpus is in cell (2,2)

  V11 = We've visited cell (1,1)

  OK11 = Cell (1,1) is safe   etc.

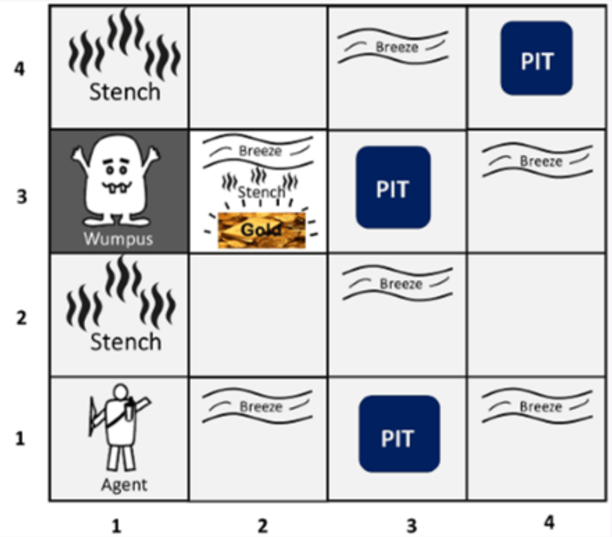
- Some rules

  (R1) $\neg S11 \rightarrow \neg W11 \land \neg W12 \land \neg W21$

  (R2) $\neg S21 \rightarrow \neg W11 \land \neg W21 \land \neg W22 \land \neg W31$

  (R3) $\neg S12 \rightarrow \neg W11 \land \neg W12 \land \neg W22 \land \neg W13$

  (R4) $S12 \rightarrow W13 \lor W12 \lor W22 \lor W11$ etc.

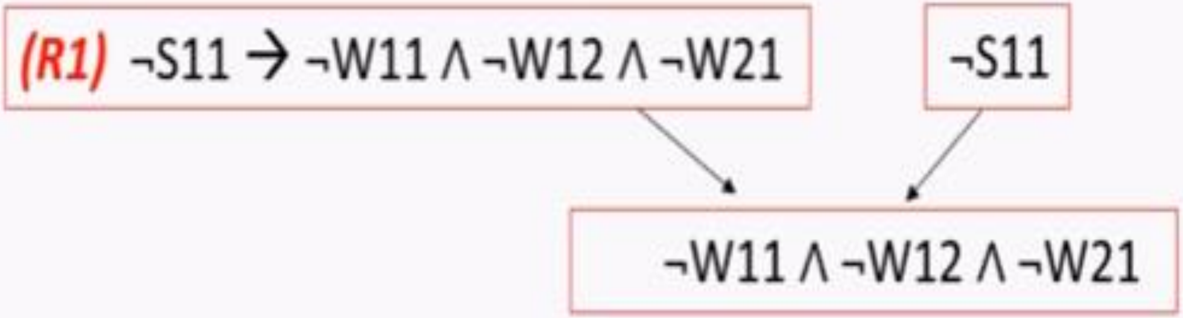


- Note that the lack of variables requires us to give similar rules for each cell.

Prove that the Wumpus is in (1,3)

Apply MP with $\neg S11$ and R1:

$\neg W11 \land \neg W12 \land \neg W21$



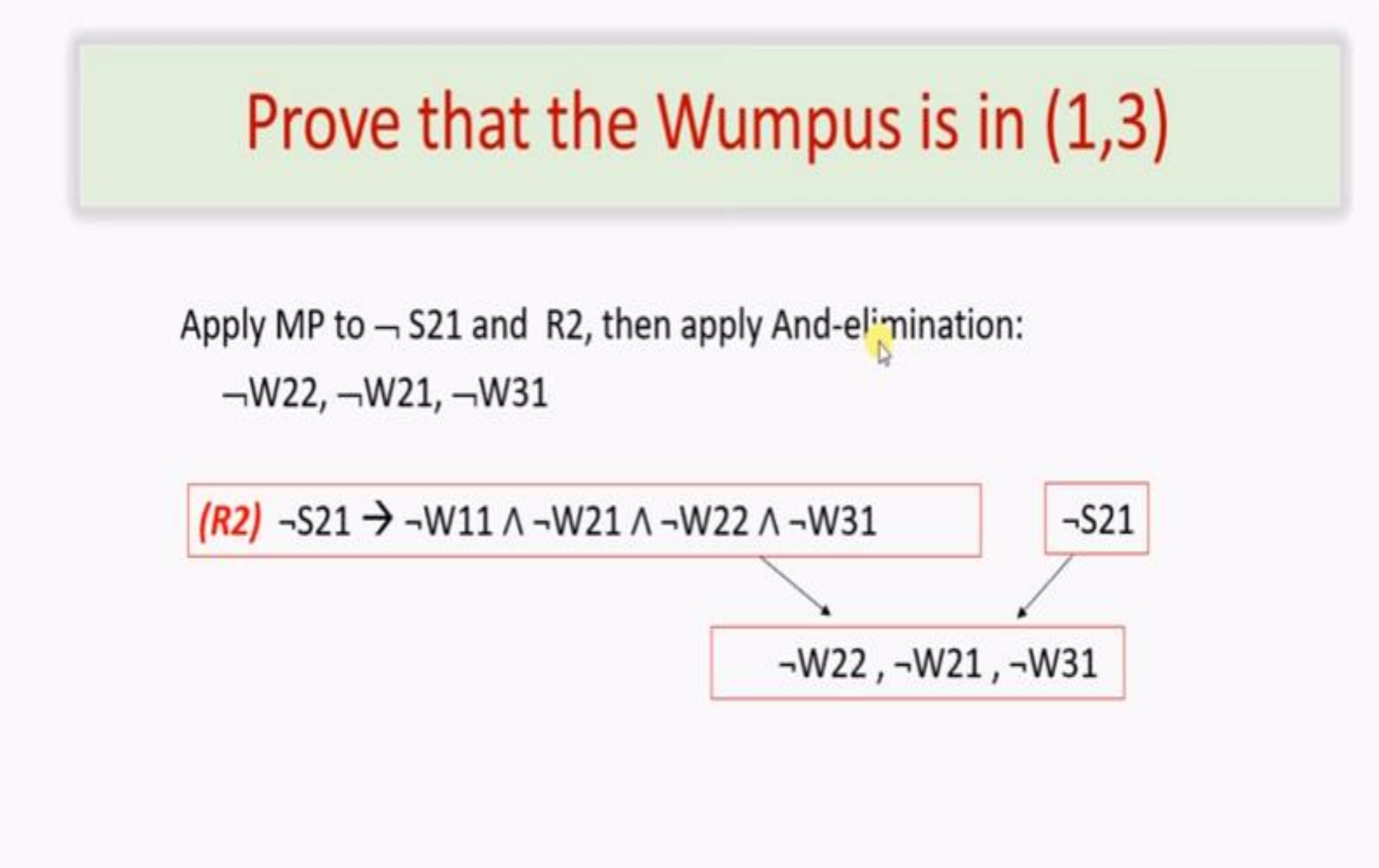
(R1) $\neg S11 \rightarrow \neg W11 \land \neg W12 \land \neg W21$

$\neg S11$

$\neg W11 \land \neg W12 \land \neg W21$

Apply And-Elimination to this we get 3 sentences:

$\neg W11, \neg W12, \neg W21$



Prove that the Wumpus is in (1,3)

Apply MP to $\neg$ S21 and R2, then apply And-elimination:

$\neg W22, \neg W21, \neg W31$

(R2) $\neg S21 \rightarrow \neg W11 \land \neg W21 \land \neg W22 \land \neg W31$

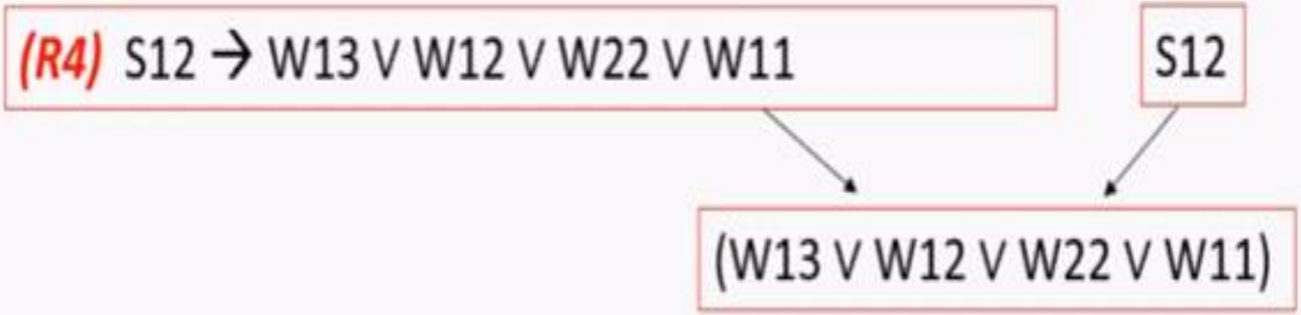
$\neg S21$

$\neg W22 , \neg W21 , \neg W31$

Prove that the Wumpus is in (1,3)

Apply MP to S12 and R4 to obtain:

W13 ∨ W12 ∨ W22 ∨ W11



(R4) S12 → W13 V W12 V W22 V W11

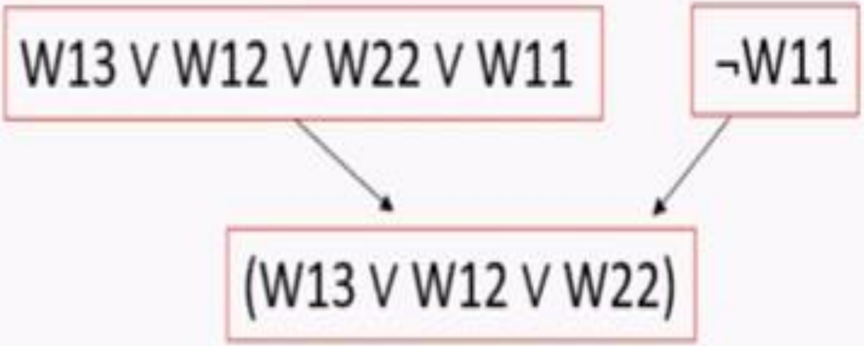
S12

(W13 V W12 V W22 V W11)

Prove that the Wumpus is in (1,3)

Apply Unit resolution on (W13 ∨ W12 ∨ W22 ∨ W11) and ¬W11:

W13 ∨ W12 ∨ W22



W13 V W12 V W22 V W11

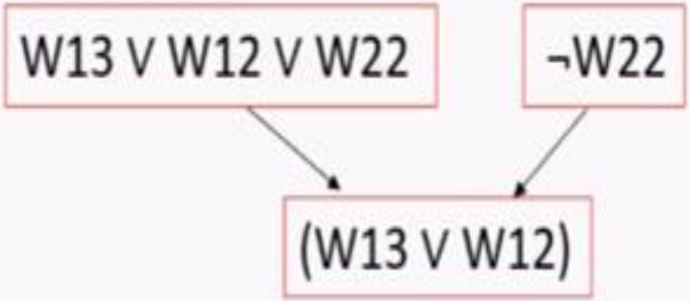
¬W11

(W13 V W12 V W22)

Prove that the Wumpus is in (1,3)

Apply Unit Resolution with (W13 ∨ W12 ∨ W22) and ¬W22:

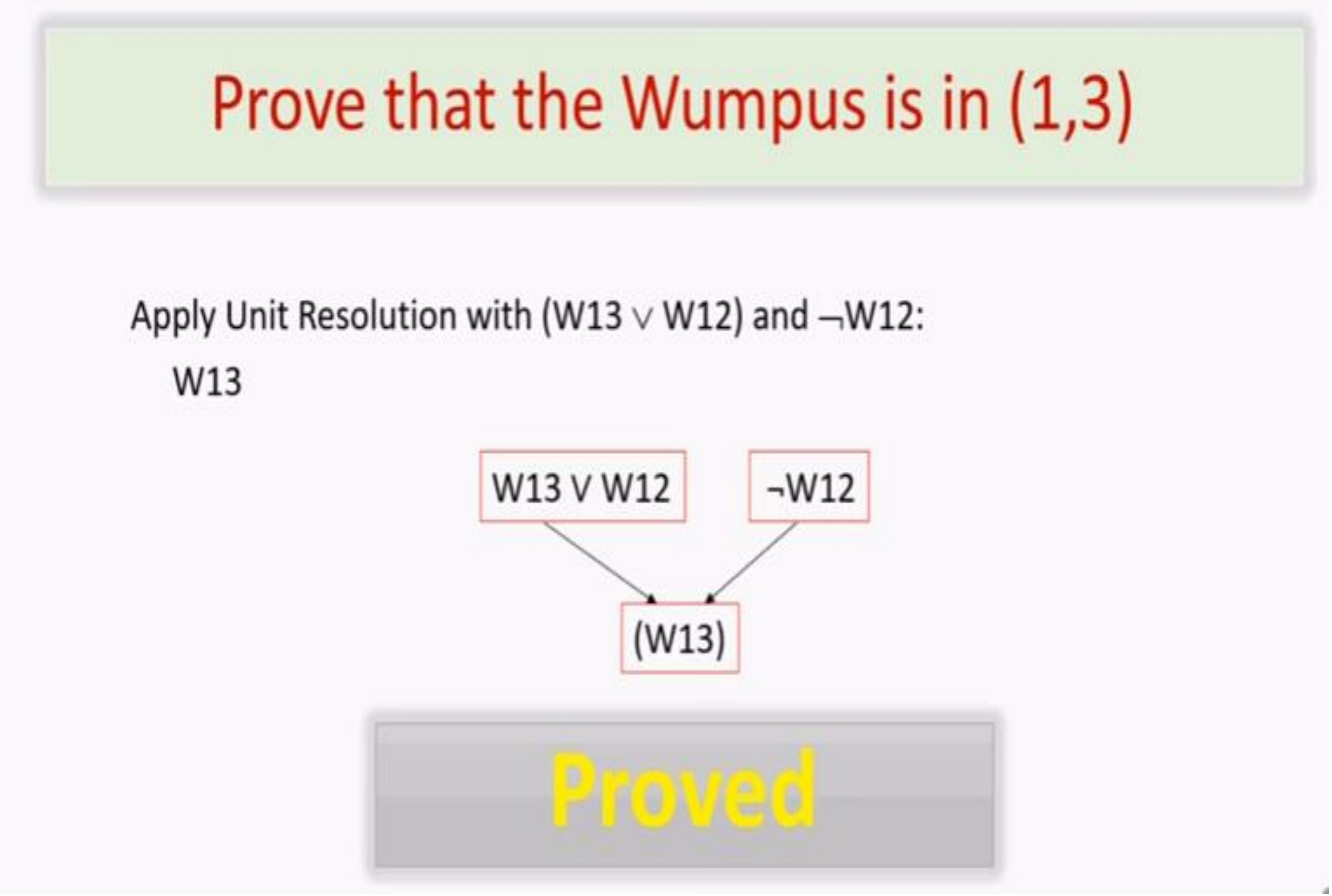
W13 ∨ W12



W13 V W12 V W22

¬W22

(W13 V W12)



Prove that the Wumpus is in (1,3)

Apply Unit Resolution with (W13 ∨ W12) and ¬W12:

W13

W13 V W12

¬W12

(W13)

Proved

Knowledge Representation or Reasoning

- Propositional Logic
- First order Logic
- Semantic Networks
- Frame representation

Logic

- Logic is concerned with the **truth of statements** about the world.
  - Generally each statement is either **TRUE or FALSE**.
  - Logic includes : **Syntax , Semantics and Inference Procedure**.
- Different logics exist, which allow you to represent different kinds of things, and which allow more or less efficient inference.
- Propositional logic, Predicate logic, Temporal logic, Modal logic, Description logic..

SYNTAX and SEMANTICS

Knowledge bases consist of sentences. (syntax and semantics)

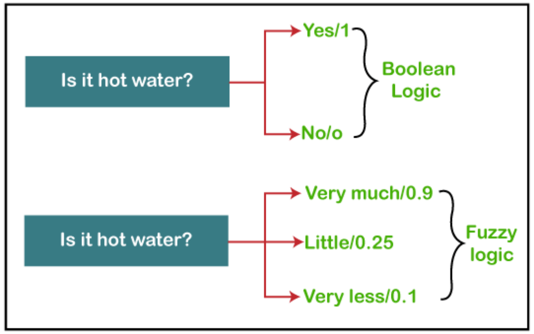
SYNTAX : **“x + y = 4” is a well-formed sentence, whereas “x4y+ =” is not**.

SEMANTICS : A logic must also define the semantics or meaning of sentences. The semantics defines TRUTH the truth of each sentence with respect to each possible world. For example, **the semantics POSSIBLE WORLD for arithmetic specifies that the sentence “x + y = 4” is true in a world where x is 2 and y is 2, but false in a world where x is 1 and y is 1**.

In standard logics, every sentence must be either true or false in each possible world—there is no “in between.

Logic

- **Types of Logic**



Is it hot water?

Yes/1

No/o

Boolean Logic

Is it hot water?

Very much/0.9

Little/0.25

Very less/0.1

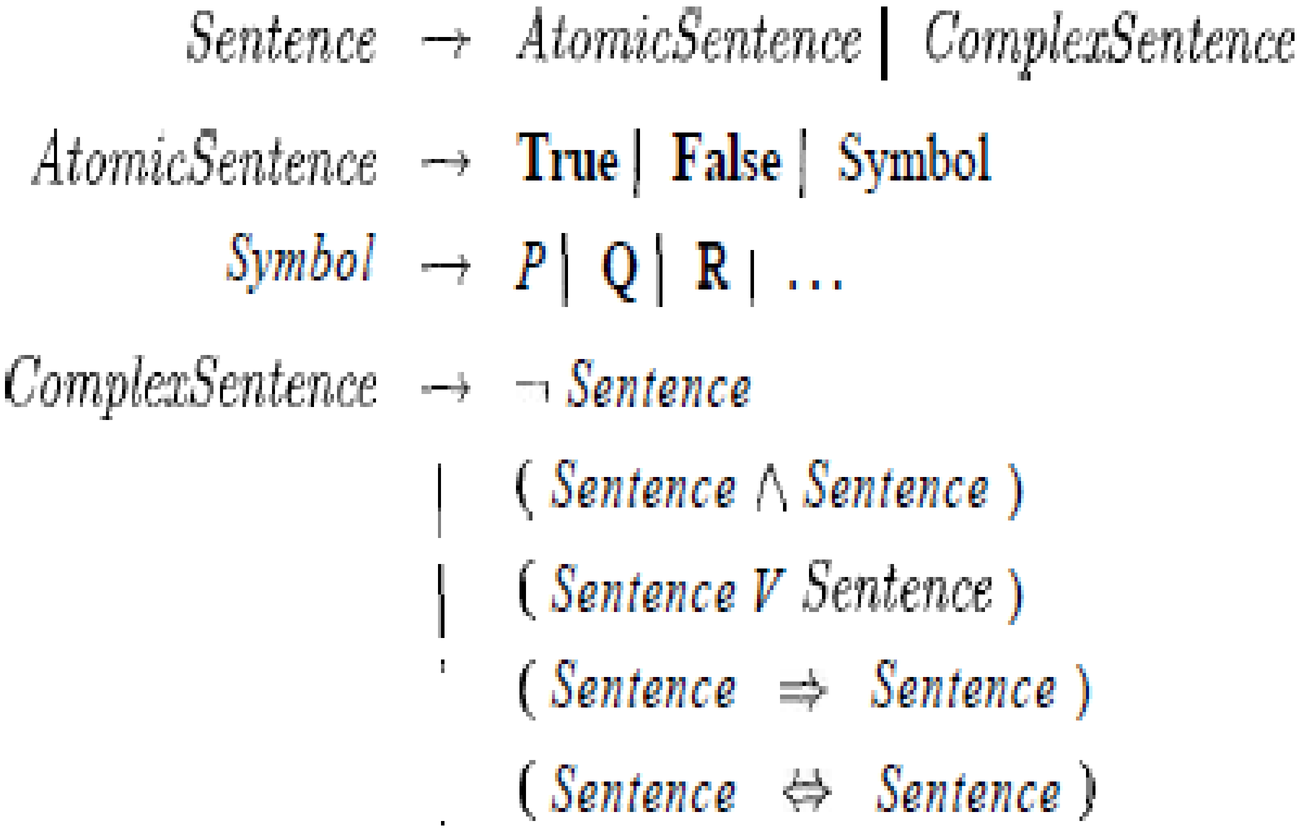
Fuzzy logic

Language	What Exists	Belief of Agent
Propositional Logic	Facts eg. “Ram is tall and he is in a football team”	T/F/Unknown
First Order Logic	Facts, objects, relations e.g. child(x,y)	T/F/Unknown
Temporal Logic	Facts, Objects, Relations, Time "I am *always* hungry", "I will *eventually* be hungry"	T/F/Unknown
Fuzzy Logic	Degree of Truth	Degree of Belief (0…1)

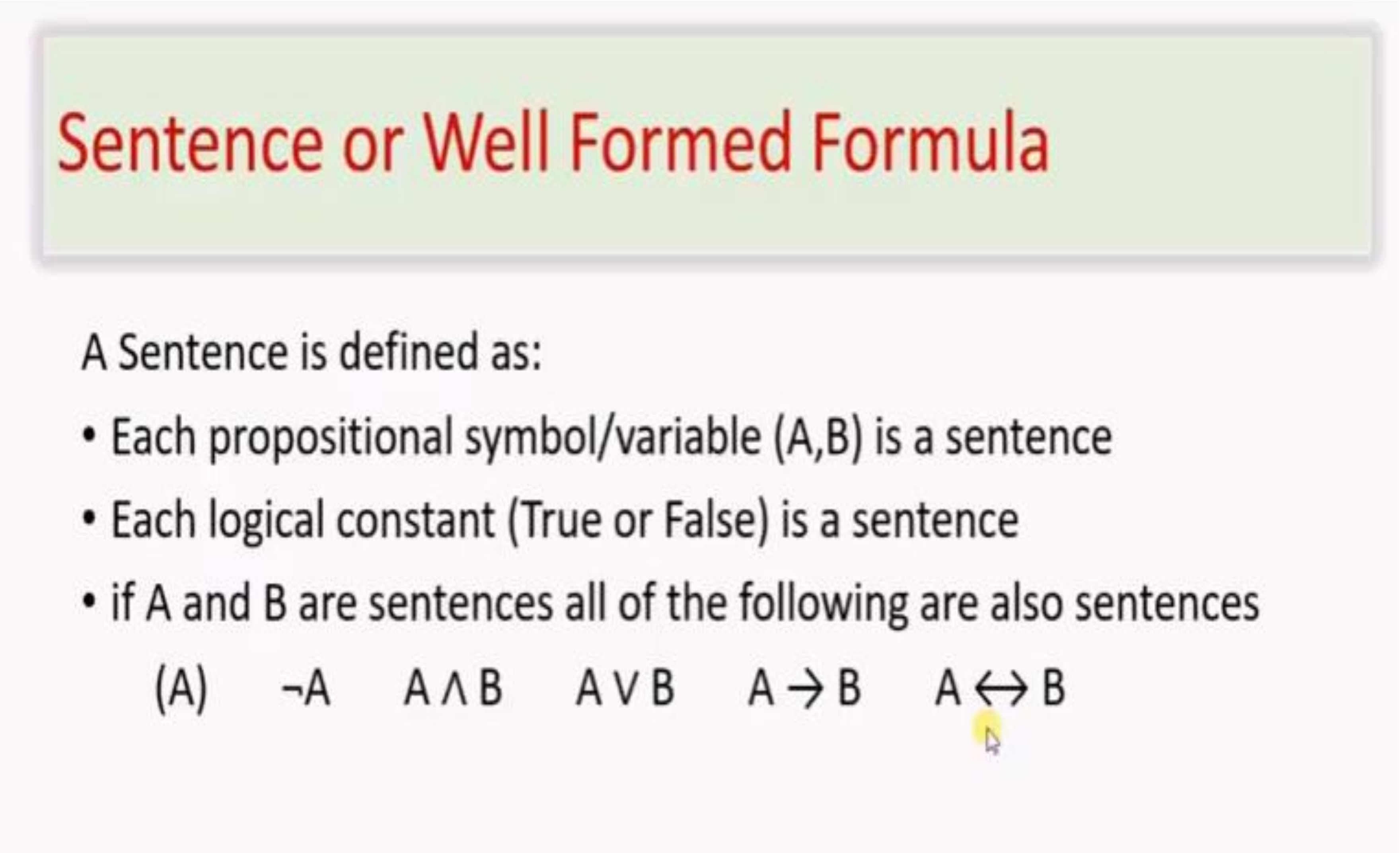
What is probability that our new manager will be fat?

Fuzzy Logic – How Fat???? - extent

Propositional Logic



$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathbf{True} \mid \mathbf{False} \mid \mathrm{Symbol} \\
\mathit{Symbol} & \rightarrow & P \mid \mathbf{Q} \mid \mathbf{R} \mid \ldots \\
\mathit{ComplexSentence} & \rightarrow & \neg\, \mathit{Sentence} \\
 & \mid & (\, \mathit{Sentence} \land \mathit{Sentence}\,) \\
 & \mid & (\, \mathit{Sentence}\ V\ \mathit{Sentence}\,) \\
 & \mid & (\, \mathit{Sentence} \Rightarrow \mathit{Sentence}\,) \\
 & \mid & (\, \mathit{Sentence} \Leftrightarrow \mathit{Sentence}\,)
\end{array}
$$

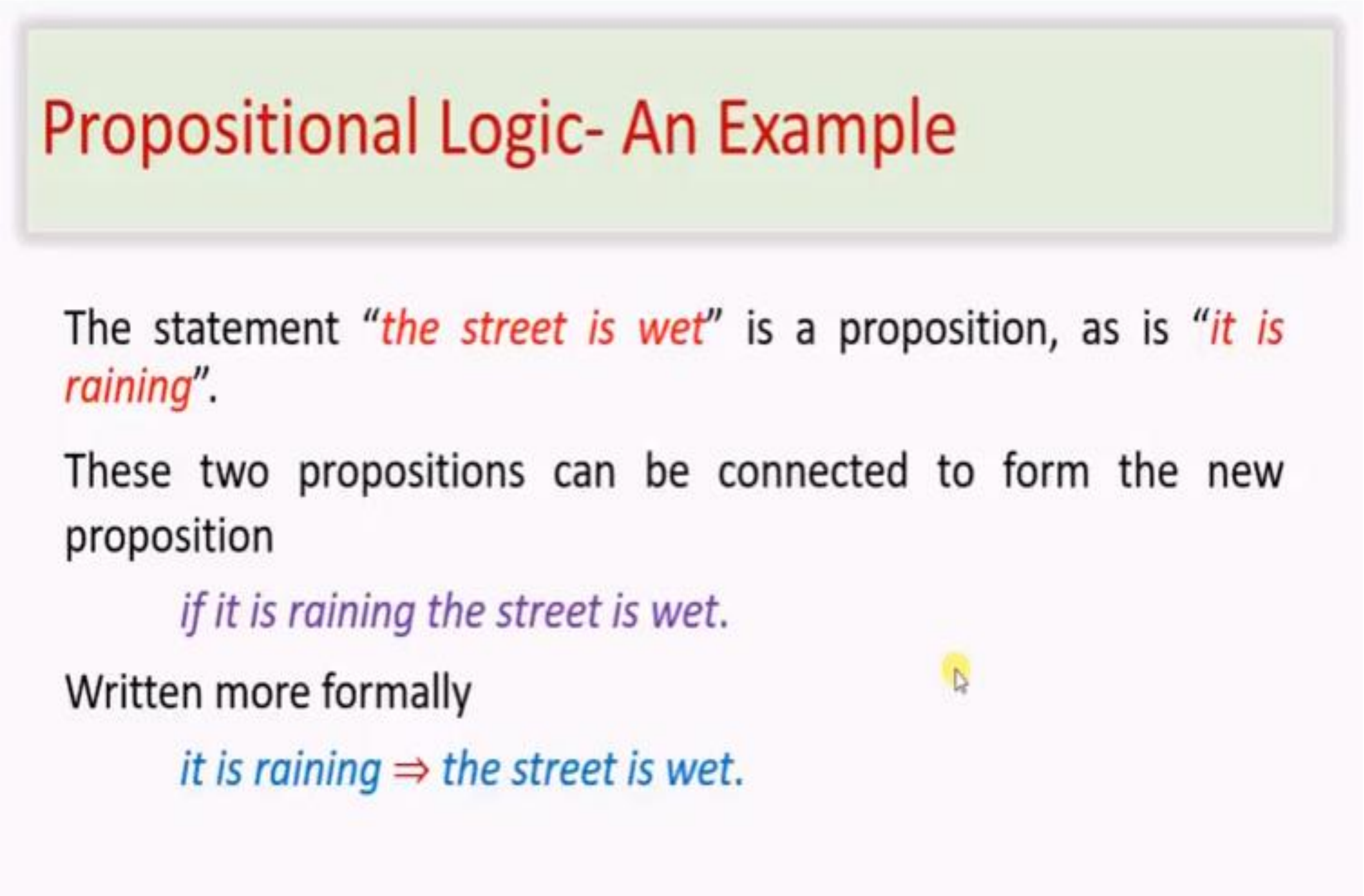


Sentence or Well Formed Formula

A Sentence is defined as:

- Each propositional symbol/variable (A,B) is a sentence
- Each logical constant (True or False) is a sentence
- if A and B are sentences all of the following are also sentences

$(A) \quad \neg A \quad A \land B \quad A \lor B \quad A \rightarrow B \quad A \leftrightarrow B$



Propositional Logic- An Example

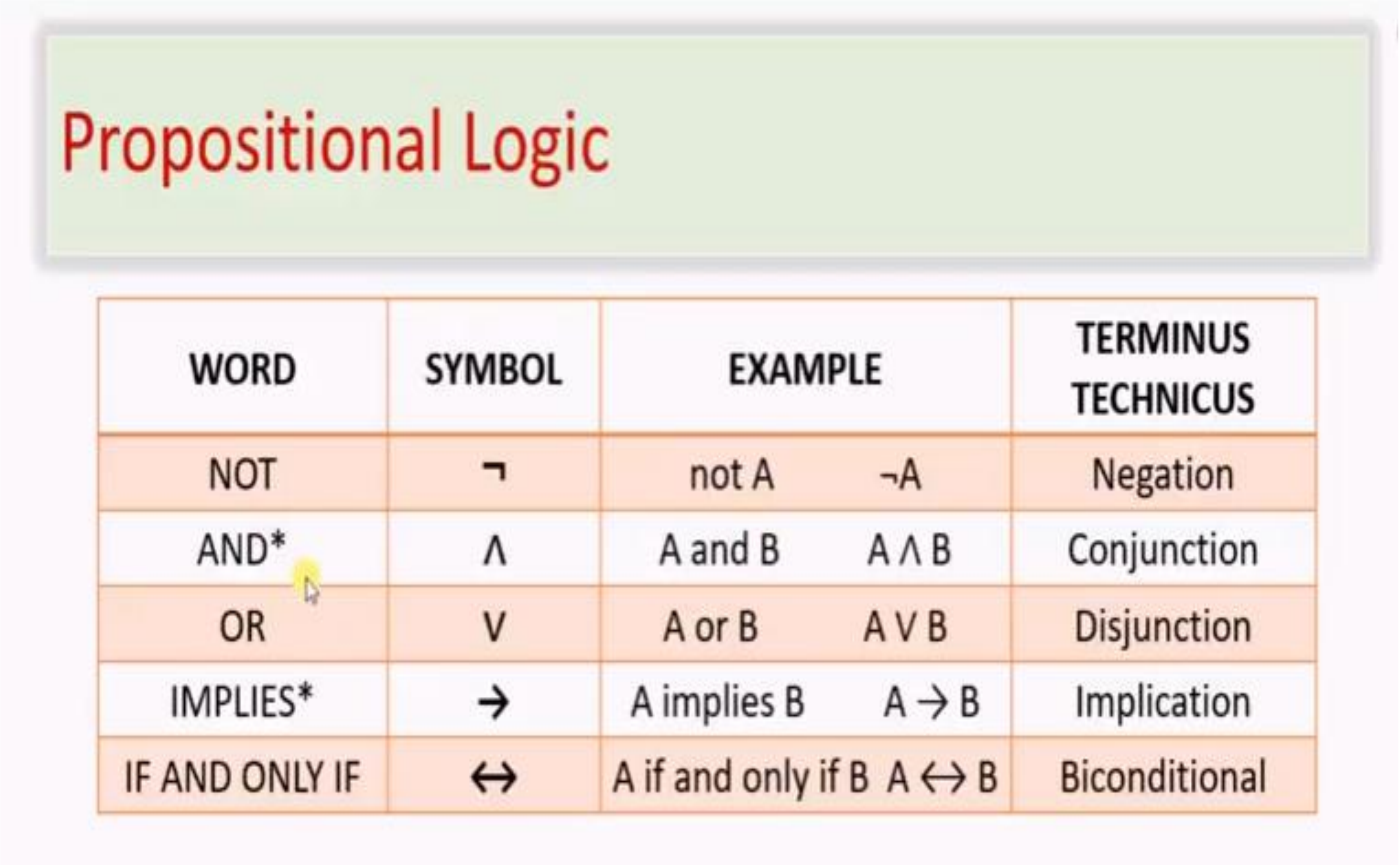
The statement "*the street is wet*" is a proposition, as is "*it is raining*".

These two propositions can be connected to form the new proposition

if it is raining the street is wet.

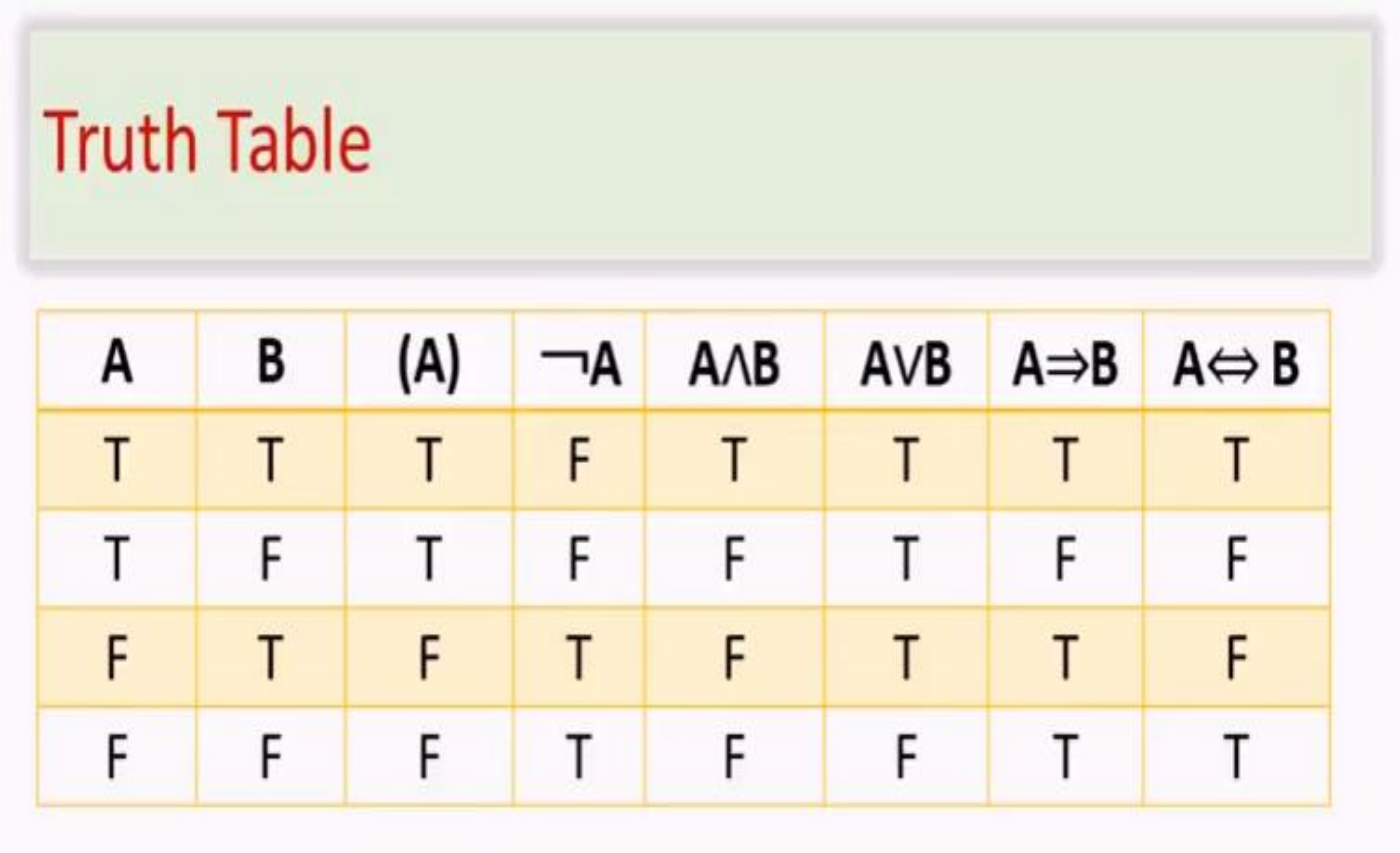
Written more formally

it is raining ⇒ *the street is wet.*



Propositional Logic

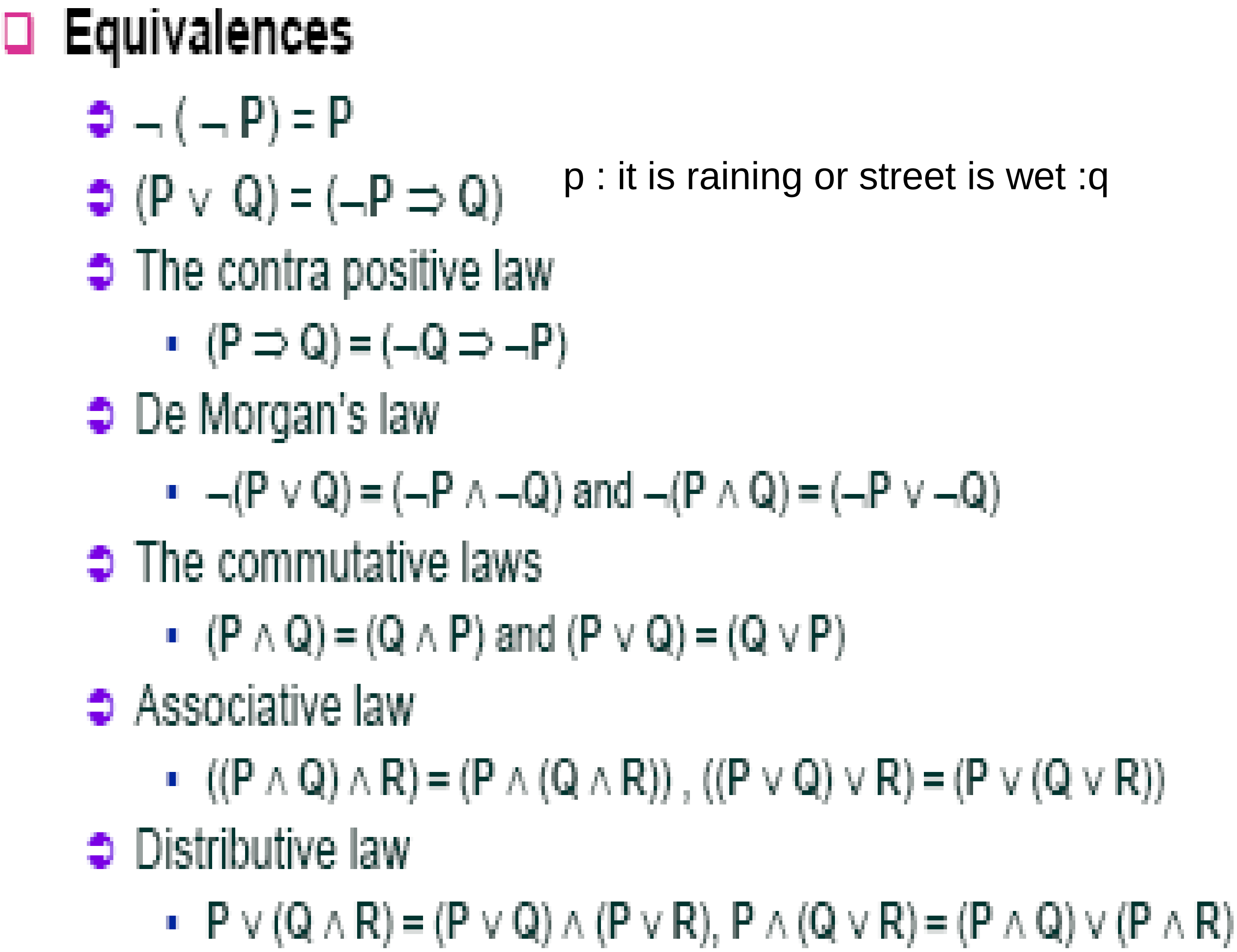
WORD	SYMBOL	EXAMPLE	TERMINUS TECHNICUS
NOT	¬	not A ¬A	Negation
AND*	∧	A and B A ∧ B	Conjunction
OR	∨	A or B A ∨ B	Disjunction
IMPLIES*	→	A implies B A → B	Implication
IF AND ONLY IF	↔	A if and only if B A ↔ B	Biconditional



Truth Table

A	B	(A)	¬A	A∧B	A∨B	A⇒B	A⇔ B
T	T	T	F	T	T	T	T
T	F	T	F	F	T	F	F
F	T	F	T	F	T	T	F
F	F	F	T	F	F	T	T

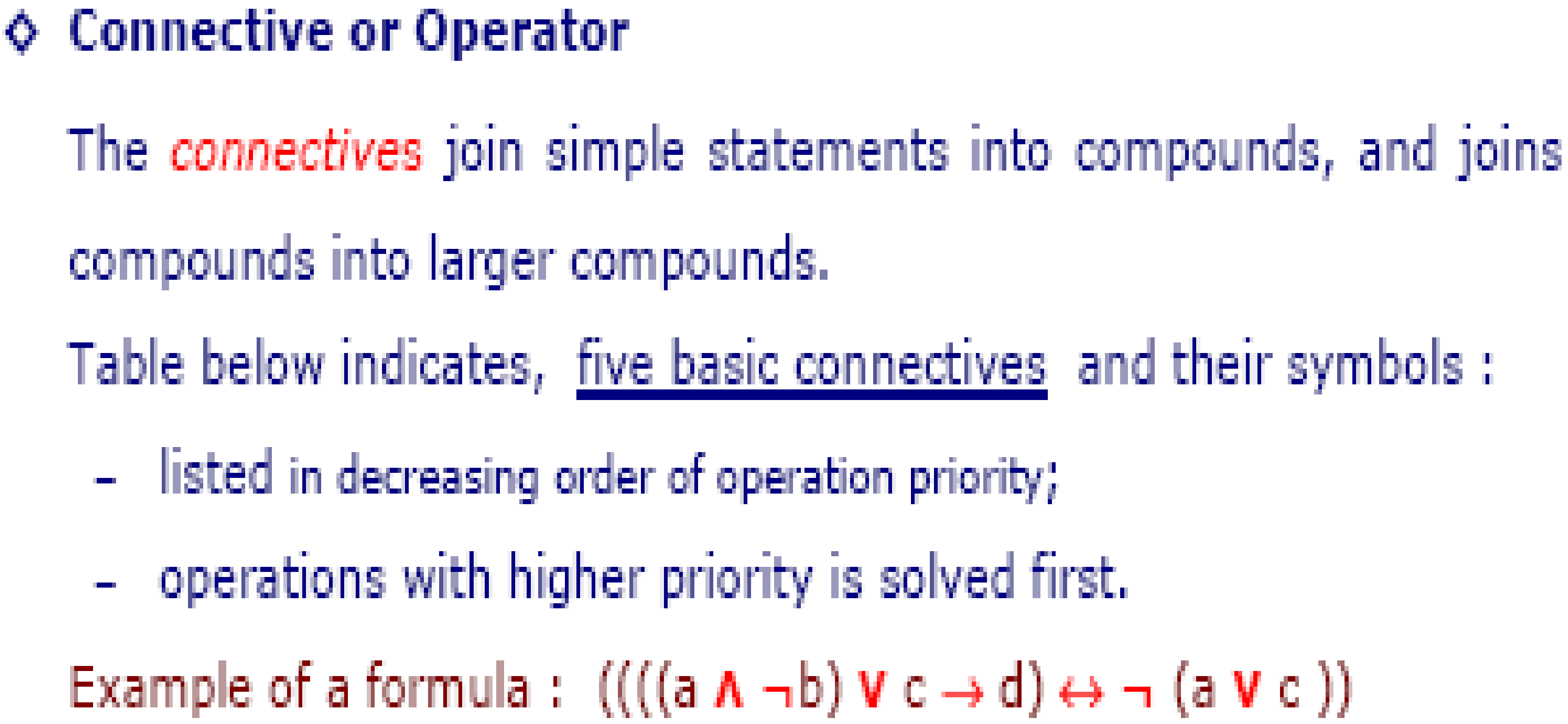
Proposition Logic : Equivalences



- Equivalences
  - $\neg(\neg P) = P$
  - $(P \vee Q) = (\neg P \supset Q)$
  - The contra positive law
    - $(P \supset Q) = (\neg Q \supset \neg P)$
  - De Morgan's law
    - $\neg(P \vee Q) = (\neg P \wedge \neg Q)$ and $\neg(P \wedge Q) = (\neg P \vee \neg Q)$
  - The commutative laws
    - $(P \wedge Q) = (Q \wedge P)$ and $(P \vee Q) = (Q \vee P)$
  - Associative law
    - $((P \wedge Q) \wedge R) = (P \wedge (Q \wedge R))$, $((P \vee Q) \vee R) = (P \vee (Q \vee R))$
  - Distributive law
    - $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$, $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$

p : it is raining or street is wet :q

Proposition Logic



◊ **Connective or Operator**

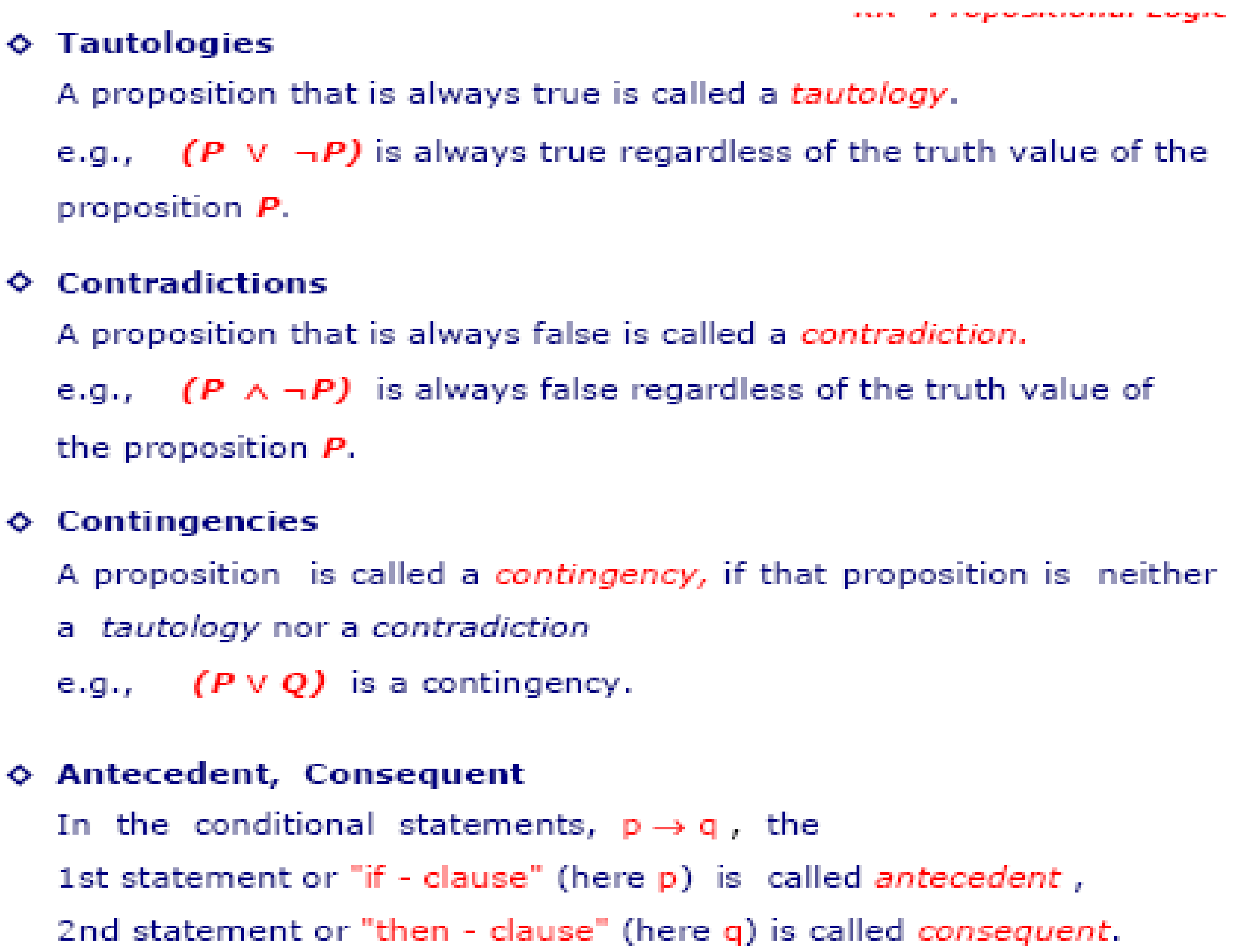
The *connectives* join simple statements into compounds, and joins compounds into larger compounds.

Table below indicates, five basic connectives and their symbols :

- listed in decreasing order of operation priority;
- operations with higher priority is solved first.

Example of a formula : $(((a \land \neg b) \lor c \rightarrow d) \leftrightarrow \neg (a \lor c))$

Proposition Logic



- **Tautologies**

  A proposition that is always true is called a *tautology*.

  e.g., $(P \lor \neg P)$ is always true regardless of the truth value of the proposition P.

- **Contradictions**

  A proposition that is always false is called a *contradiction*.

  e.g., $(P \land \neg P)$ is always false regardless of the truth value of the proposition P.

- **Contingencies**

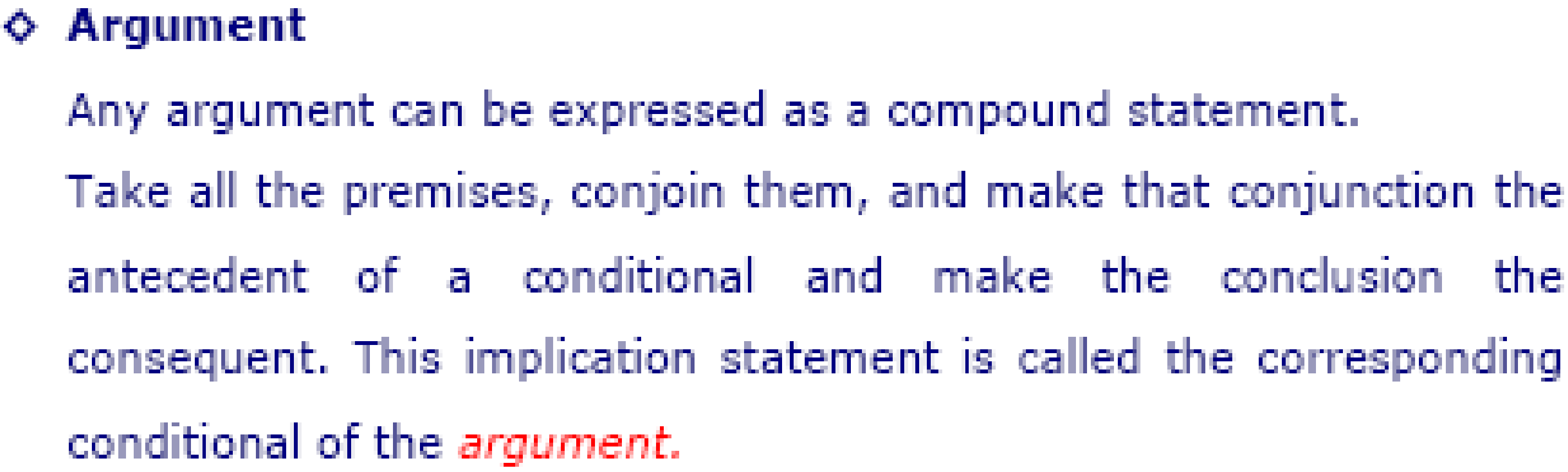
  A proposition is called a *contingency*, if that proposition is neither a *tautology* nor a *contradiction*

  e.g., $(P \lor Q)$ is a contingency.

- **Antecedent, Consequent**

  In the conditional statements, $p \to q$, the 1st statement or "if - clause" (here p) is called *antecedent* , 2nd statement or "then - clause" (here q) is called *consequent*.

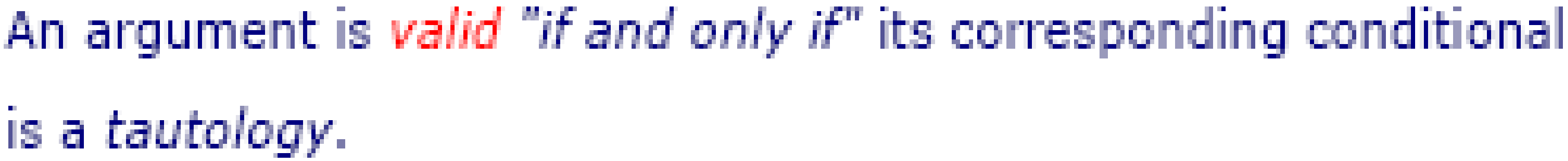
Proposition Logic



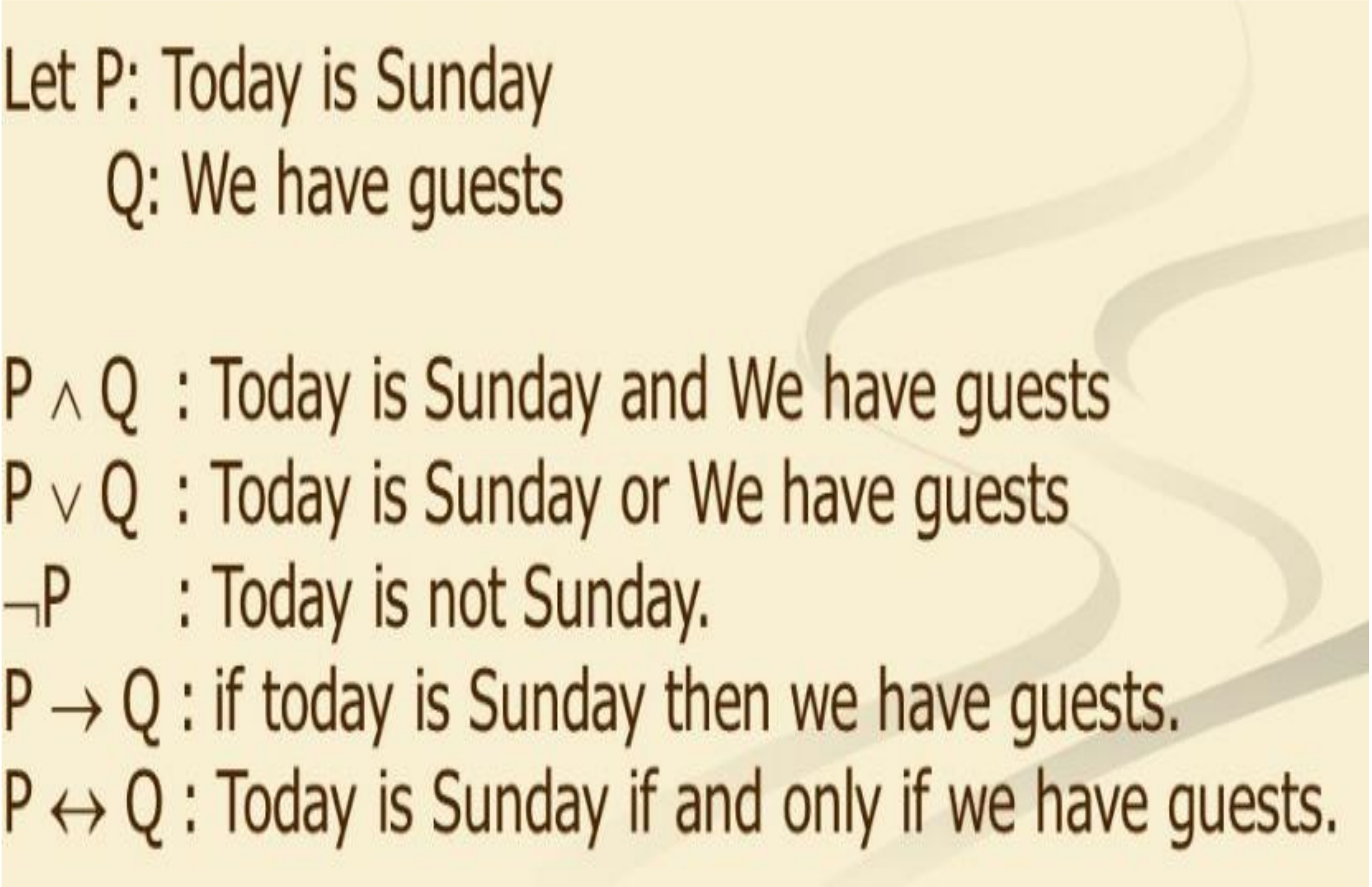
- **Argument**

  Any argument can be expressed as a compound statement.

  Take all the premises, conjoin them, and make that conjunction the antecedent of a conditional and make the conclusion the consequent. This implication statement is called the corresponding conditional of the *argument.*

‡ 

An argument is *valid "if and only if"* its corresponding conditional is a *tautology*.



Let P: Today is Sunday

Q: We have guests

$P \wedge Q$: Today is Sunday and We have guests

$P \vee Q$: Today is Sunday or We have guests

$\neg P$: Today is not Sunday.

$P \rightarrow Q$: if today is Sunday then we have guests.

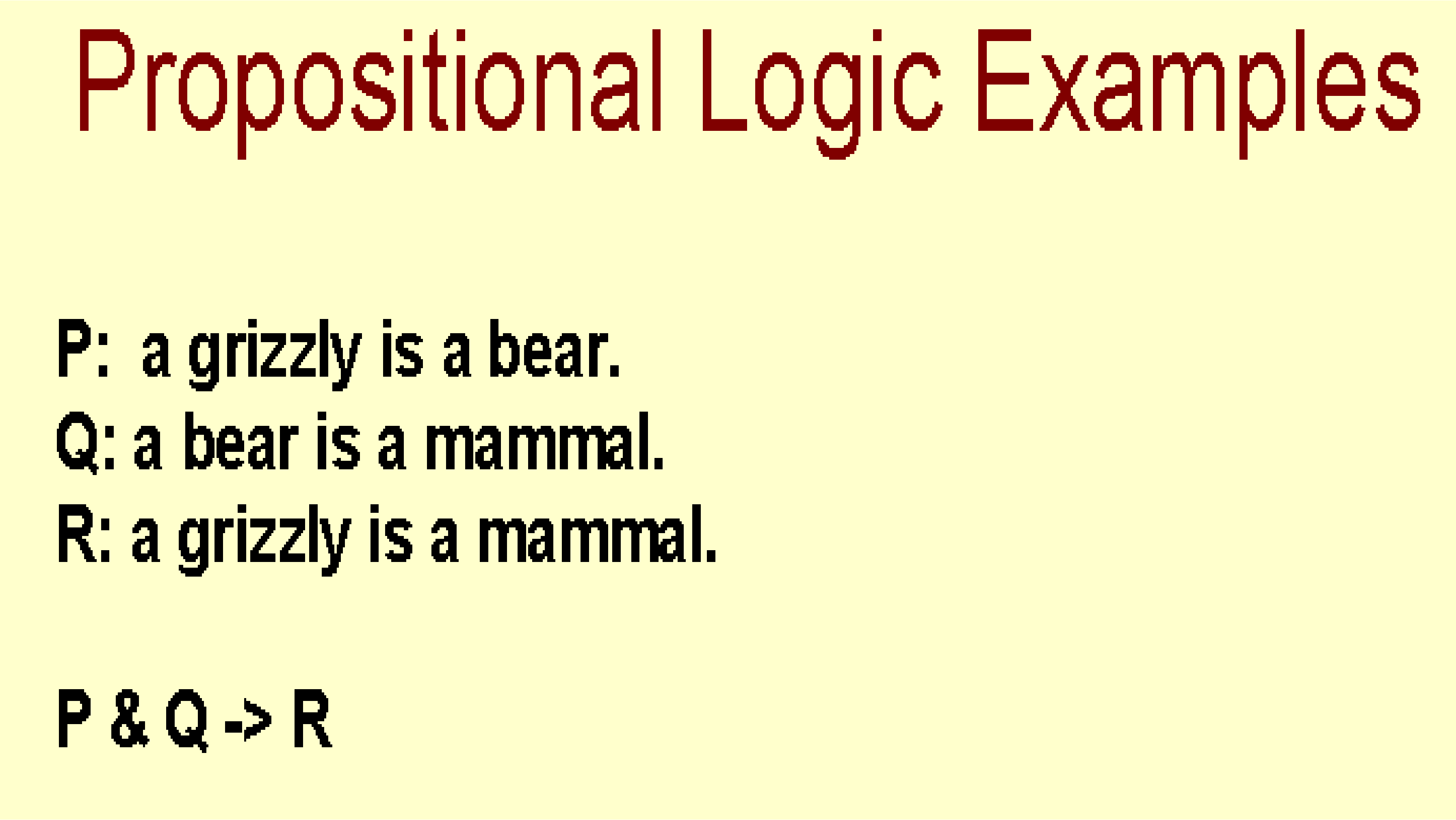
$P \leftrightarrow Q$: Today is Sunday if and only if we have guests.

Propositional Logic Examples

P: a grizzly is a bear.

Q: a bear is a mammal.

R: a grizzly is a mammal.



Propositional Logic Examples

P: a grizzly is a bear.

Q: a bear is a mammal.

R: a grizzly is a mammal.

P & Q -> R

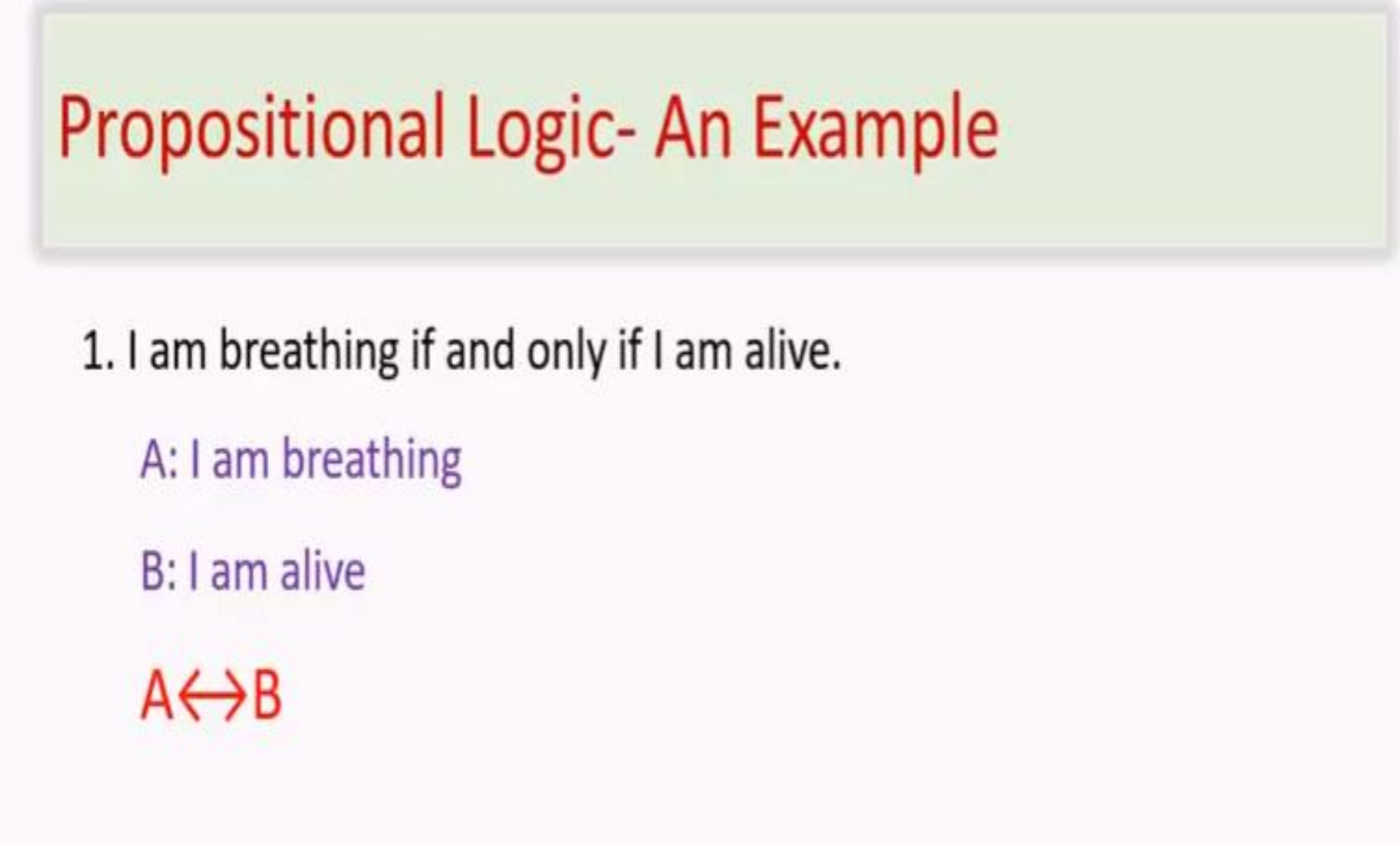
Propositional Logic- An Example

1. It is Cold and Dark.

   A: It is Cold

   B: It is Dark

   $A \wedge B$



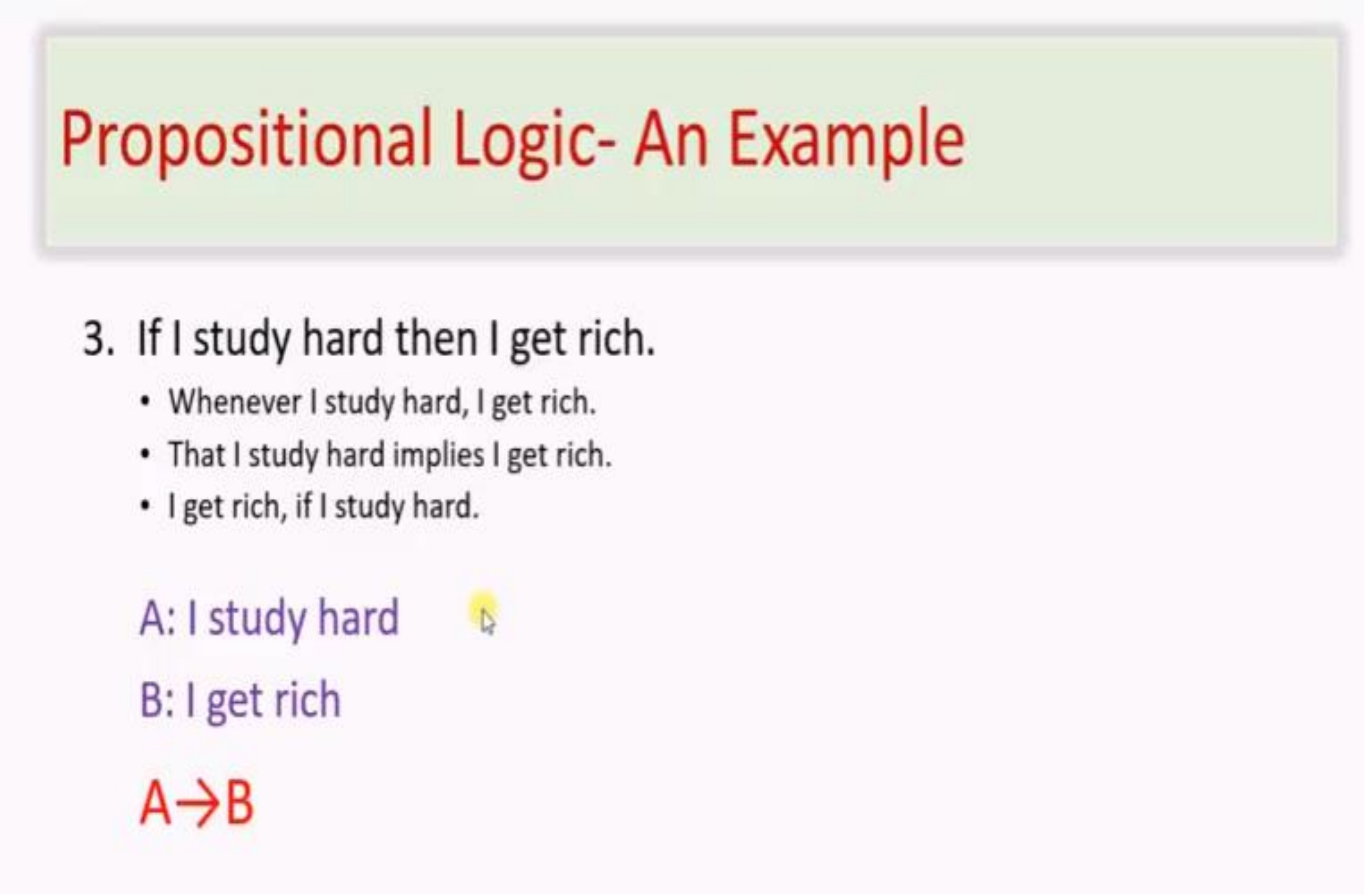
Propositional Logic- An Example

1. I am breathing if and only if I am alive.

A: I am breathing

B: I am alive

$A \leftrightarrow B$



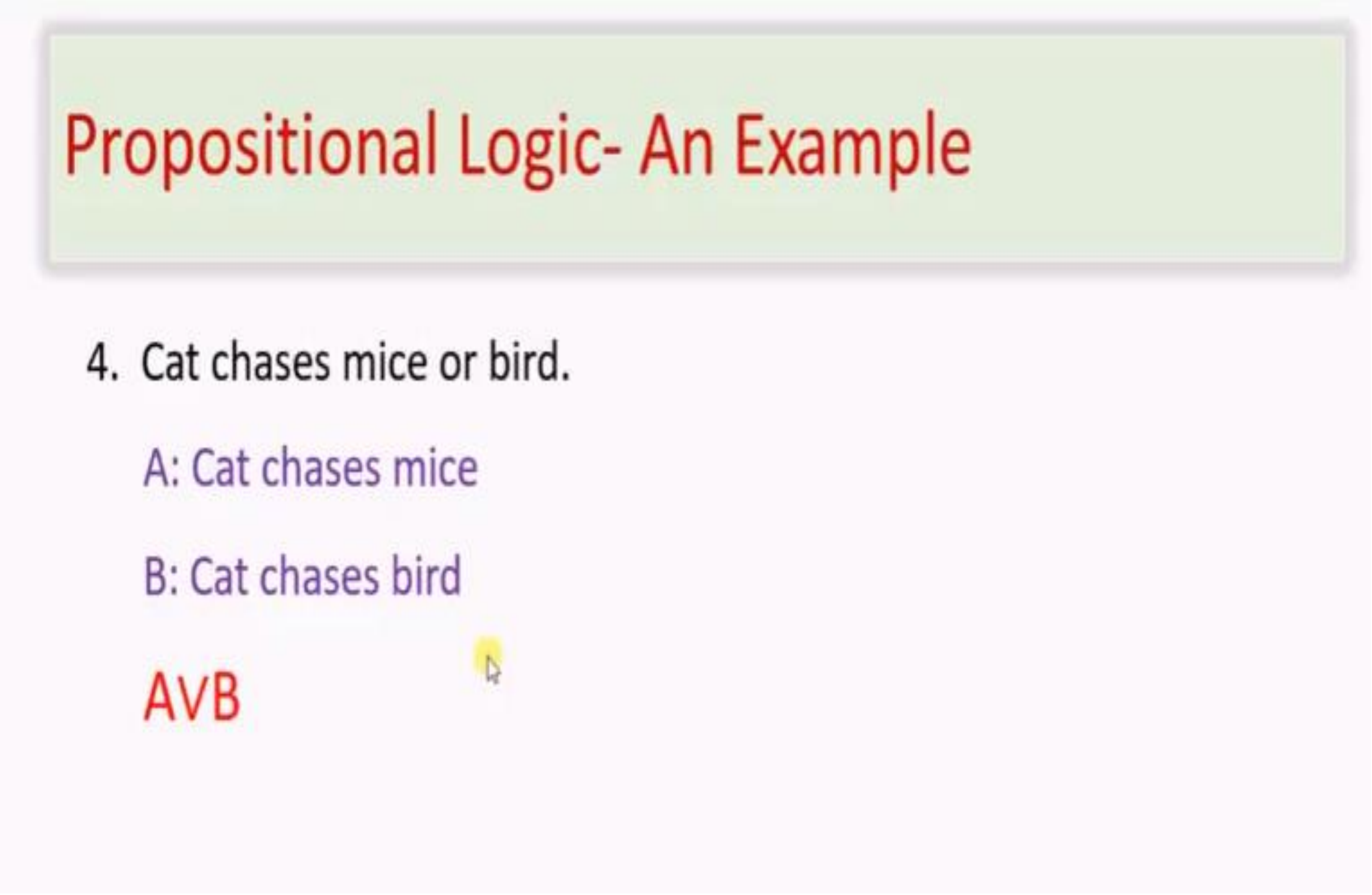
Propositional Logic- An Example

3. If I study hard then I get rich.
   - Whenever I study hard, I get rich.
   - That I study hard implies I get rich.
   - I get rich, if I study hard.

A: I study hard

B: I get rich

A→B



Propositional Logic- An Example

4. Cat chases mice or bird.

A: Cat chases mice

B: Cat chases bird

AVB

Propositional Logic- An Example

5. Logic is not easy.

A: Logic is [redacted] easy

¬A

Express following sentences in propositional logic

- It rains in July
  - P: It rains in July
- The book is not costly
  - Q: The book is costly **~(Q)**
- If it rains today and one does not carry umbrella he will be drenched

  p: It rains today q: one carry umbrella r: he will be drenched

  (p ^ ~q)□ r
- If anil is intelligent and anil is hardworking then anil scores high marks

  P: Anil is intelligent q: anil is hardworking r: anil scores high marks

  p ^ q □ r
- If the requirement of computer engineers is increased, then more seats will be offered by University and more computers will be purchased by the university computer department if the rates are compatible.

  p: The requirement of Computer engineers is increased

  q: More seats will be offered by University

  r: More computers will be purchased by the university computer department

  s: The rates are compatible

  p□ (q ^ (s□ r))

Exercise

- Let *p*, *q*, and *r* be the following propositions:
- ***p*:** You get an A on the final exam
- ***q*:** You do every exercise in the book.
- ***r*:** You get an A in this class.
- Write the following formulas using *p*, *q*, and *r* and logical connectives.
- You get an A in this class, but you do not do every exercise in the book.?
- To get an A in this class, it is necessary for you to get an A on the final.?
- Getting an A on the final and doing every exercise in the book is sufficient for getting an A in this class.?

Solution

- $r \wedge \neg q$
- $r \Rightarrow p$
- $p \wedge q \Rightarrow r$

Given the following symbols and sentences:
C to indicate that Gianni is a climber;
F to indicate that Gianni is fit;
L to indicate that Gianni is lucky:
E to indicate that Gianni climbs mount Everest.

If Gianni is a climber and he is fit, he climbs mount Everest.
If Gianni is not lucky and he is not fit, he does not climb mount Everest.
Gianni is fit.

(a) Formalize the above sentences in propositional logic.

Given the following symbols and sentences:
C to indicate that Gianni is a climber;
F to indicate that Gianni is fit;
L to indicate that Gianni is lucky:
E to indicate that Gianni climbs mount Everest.

If Gianni is a climber and he is fit, he climbs mount Everest.
If Gianni is not lucky and he is not fit, he does not climb mount Everest.
Gianni is fit.

(a)Formalize the above sentences in propositional logic.

$(C \land F) \rightarrow E$
$(\neg L \land \neg F) \rightarrow \neg E$
F

KNOWLEDGE AND REASONING

FOL (PREDICATE LOGIC) – PART II

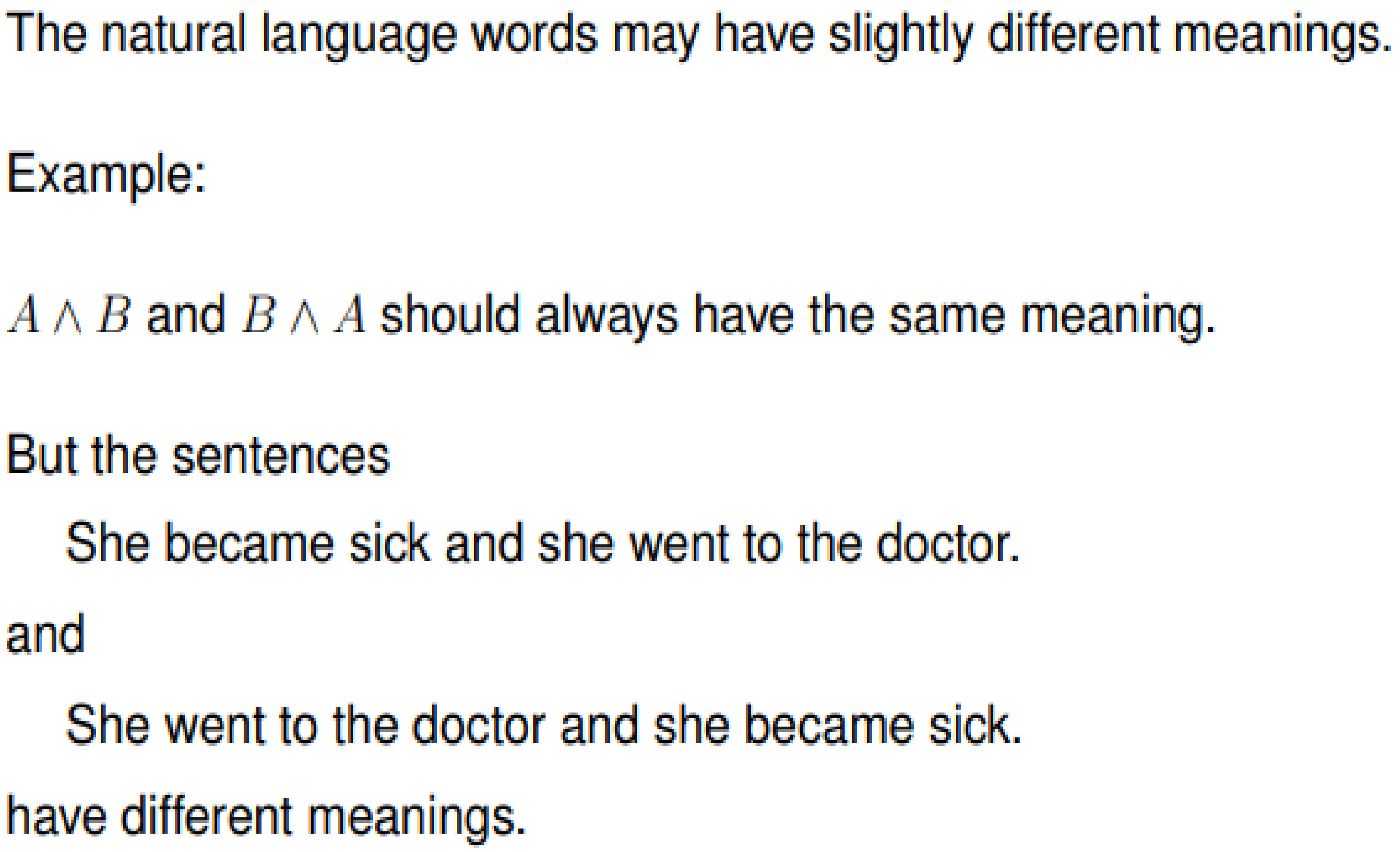
Propositional logic is a weak language

- Hard to identify "individuals" (e.g., Mary, 3)
- Can't directly talk about properties of individuals or relations between individuals (e.g., "Bill is tall")
- Generalizations, patterns, regularities can't easily be represented (e.g., "all triangles have 3 sides")
- First-Order Logic (abbreviated FOL) is expressive enough to concisely represent this kind of information

FOL adds relations, variables, and quantifiers, e.g.,

- *"Every elephant is gray":* $\forall$ x (elephant(x) → gray(x))
- *"There is a white alligator":* $\exists$ x (alligator(X) ^ white(X))

PL : DEMERITS



The natural language words may have slightly different meanings.

Example:

$A \wedge B$ and $B \wedge A$ should always have the same meaning.

But the sentences

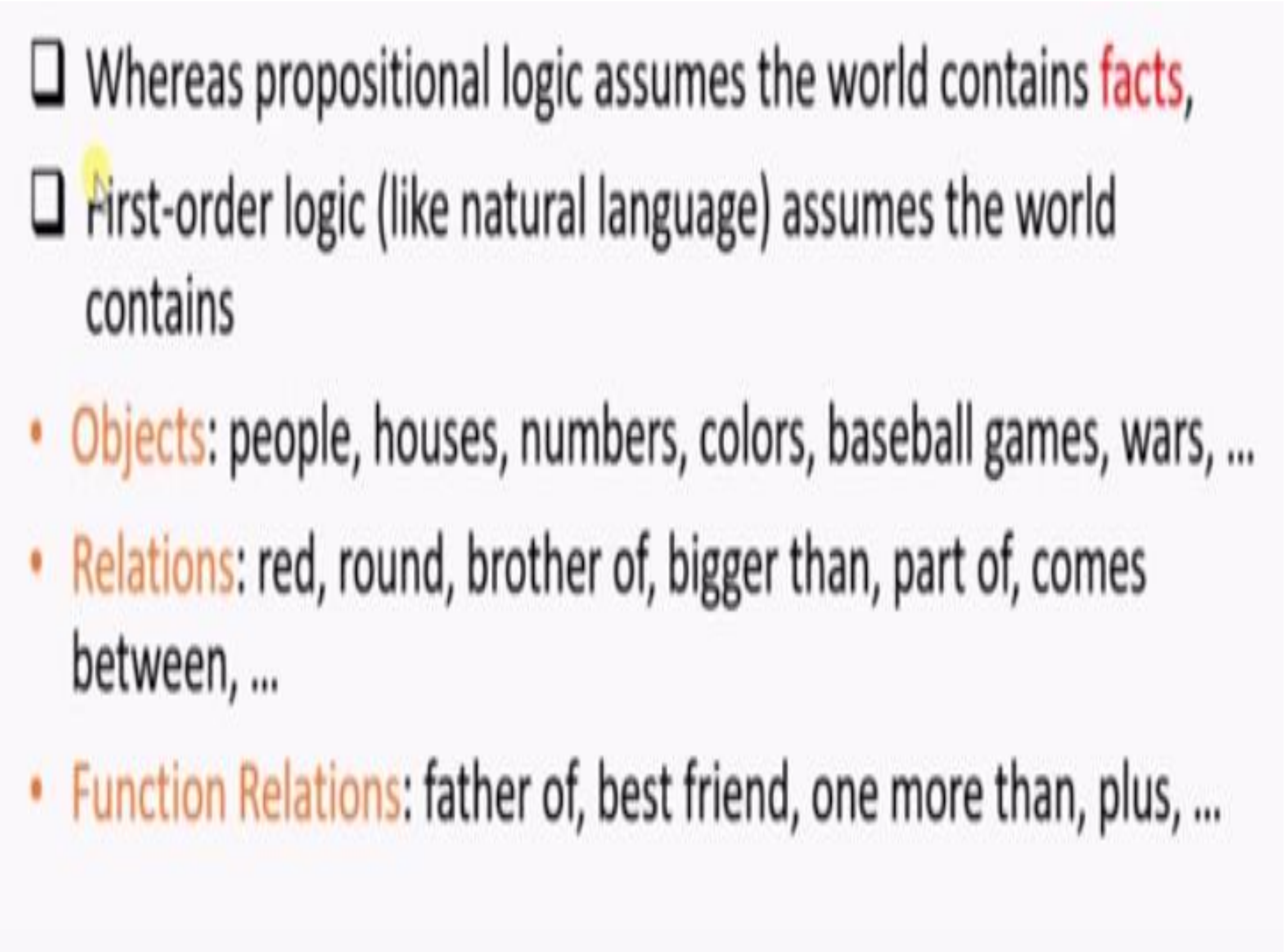
She became sick and she went to the doctor.

and

She went to the doctor and she became sick.

have different meanings.

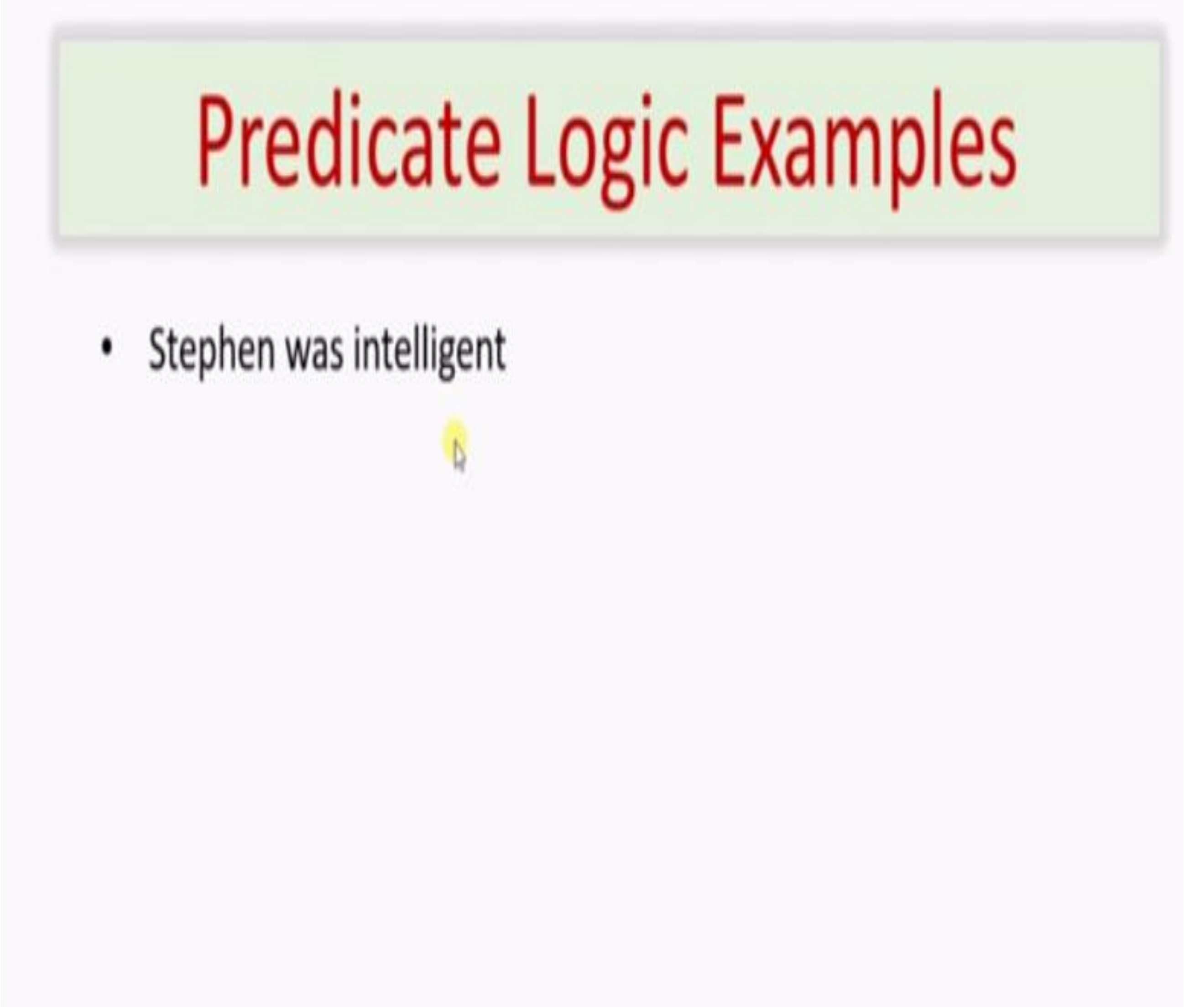
FIRST ORDER PREDICATE LOGIC



□ Whereas propositional logic assumes the world contains facts,

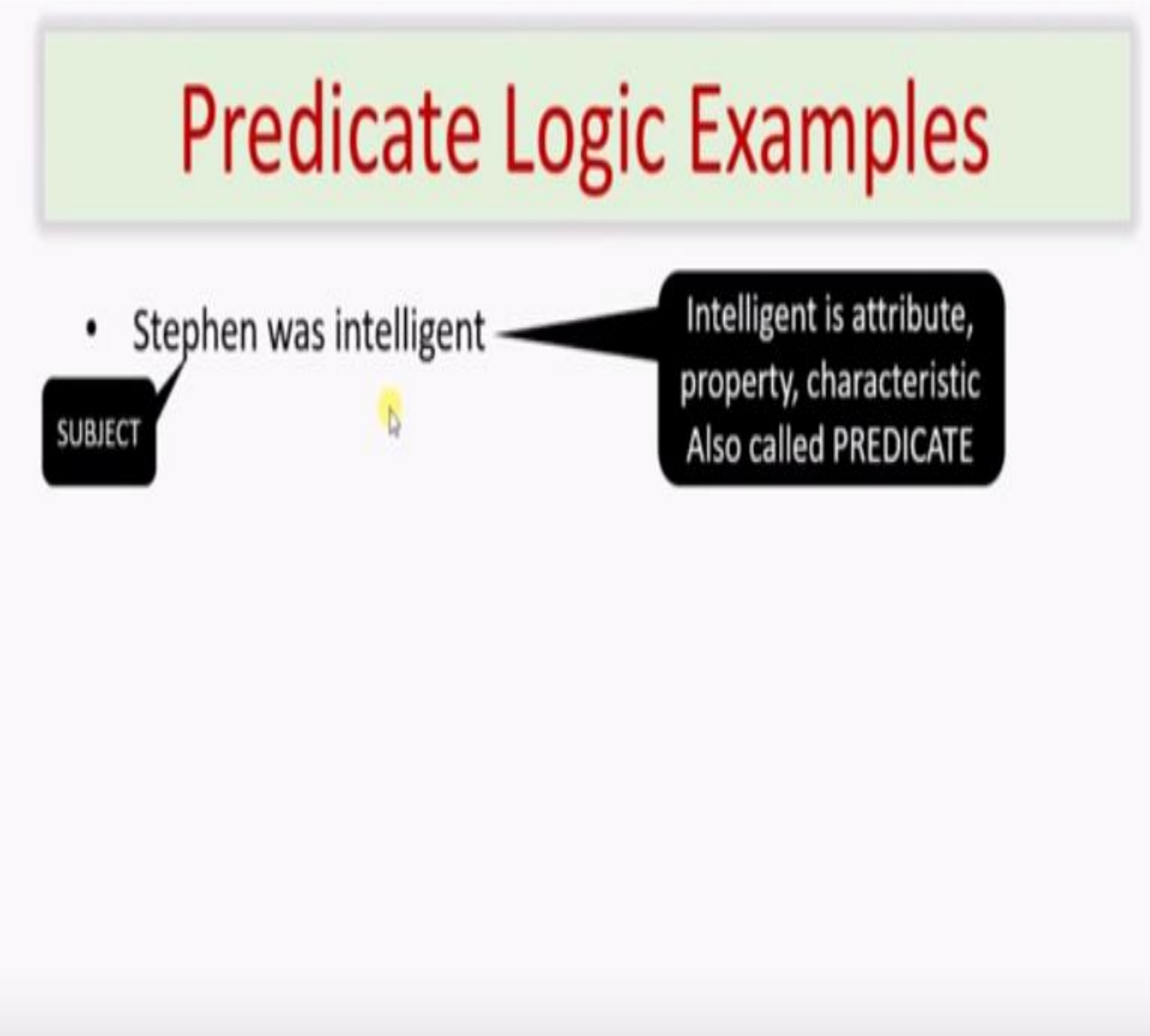
□ First-order logic (like natural language) assumes the world contains

- Objects: people, houses, numbers, colors, baseball games, wars, ...
- Relations: red, round, brother of, bigger than, part of, comes between, ...
- Function Relations: father of, best friend, one more than, plus, ...



Predicate Logic Examples

- Stephen was intelligent

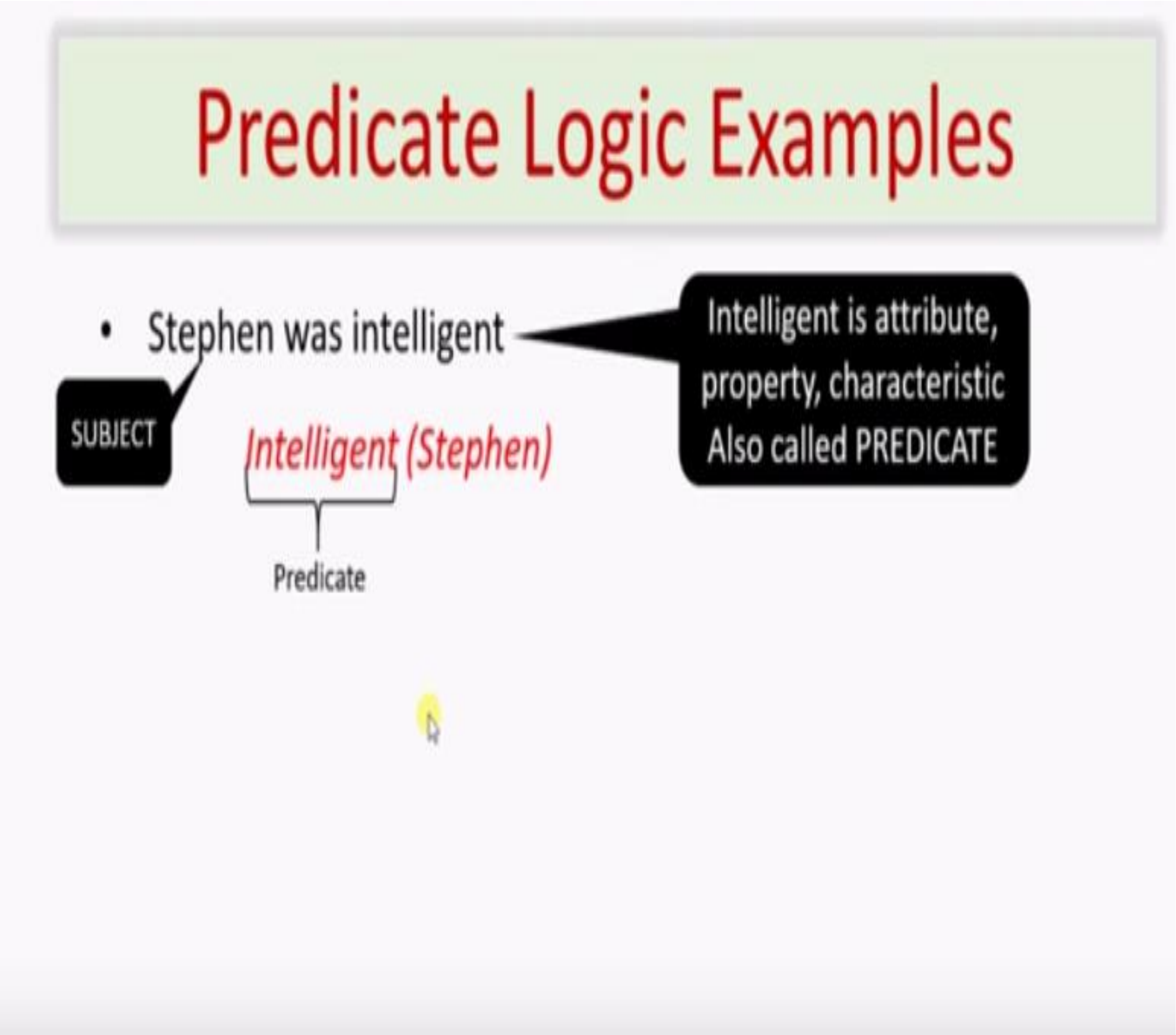


Predicate Logic Examples

- Stephen was intelligent

SUBJECT

Intelligent is attribute, property, characteristic Also called PREDICATE



Predicate Logic Examples

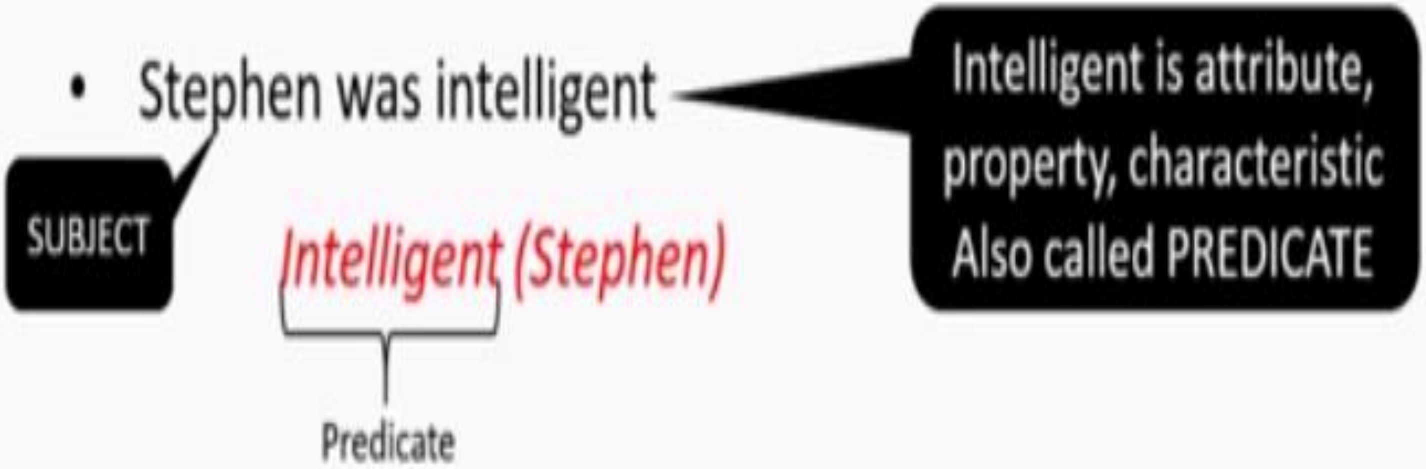
- Stephen was intelligent

SUBJECT

Intelligent is attribute, property, characteristic Also called PREDICATE

$\underbrace{\textit{Intelligent}}_{\text{Predicate}}\ \textit{(Stephen)}$

Predicate Logic Examples



- Stephen was intelligent

SUBJECT

Intelligent (Stephen)

Predicate

Intelligent is attribute, property, characteristic Also called PREDICATE

- Amir and Shahrukh are rivals

rivals(Amir,Shahrukh)

PREDICATE LOGIC EXAMPLES

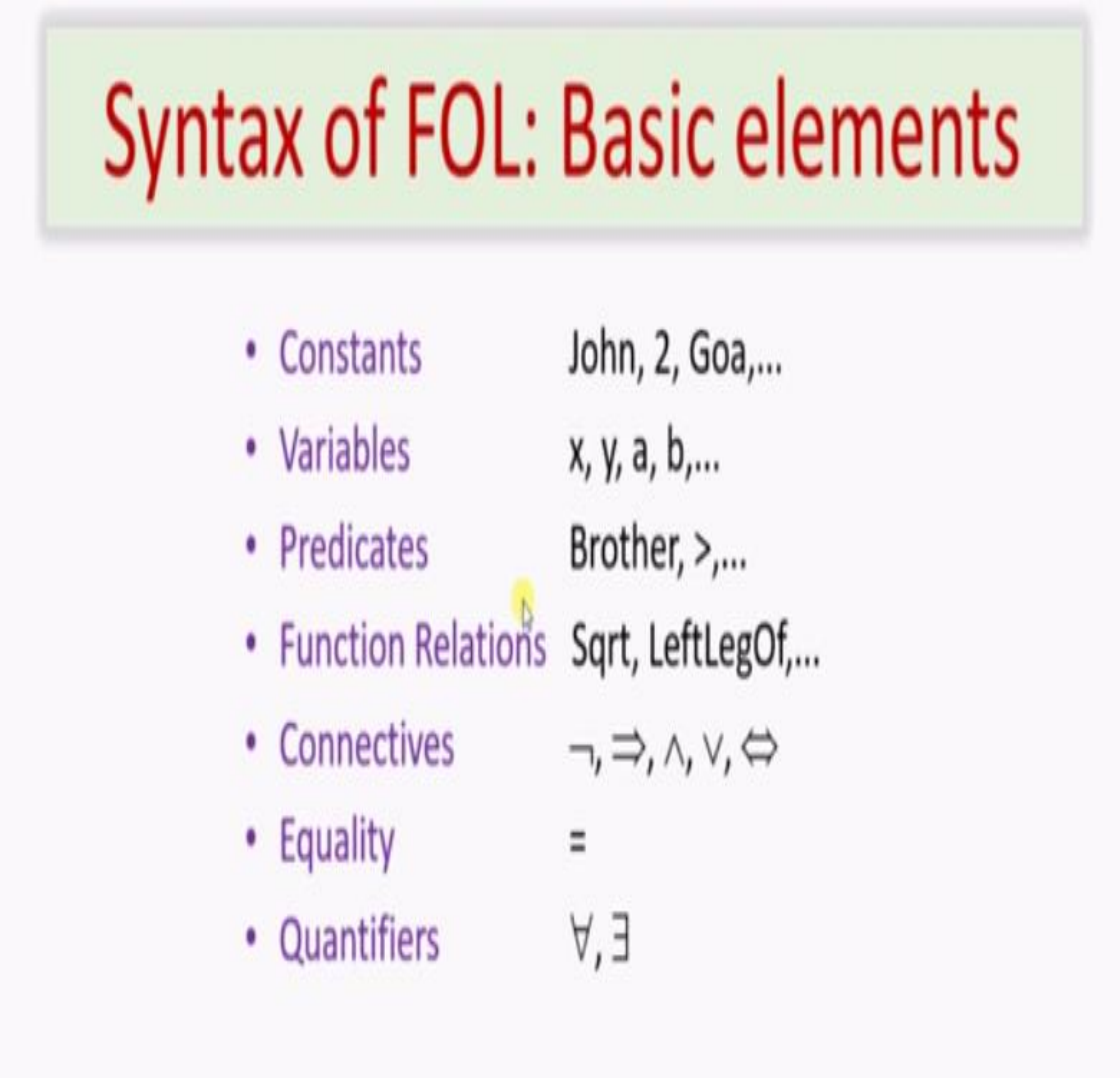
- RITAS FATHER IS MARRIED TO JOHNS MOTHER

MARRIED()

FATHER(RITA)

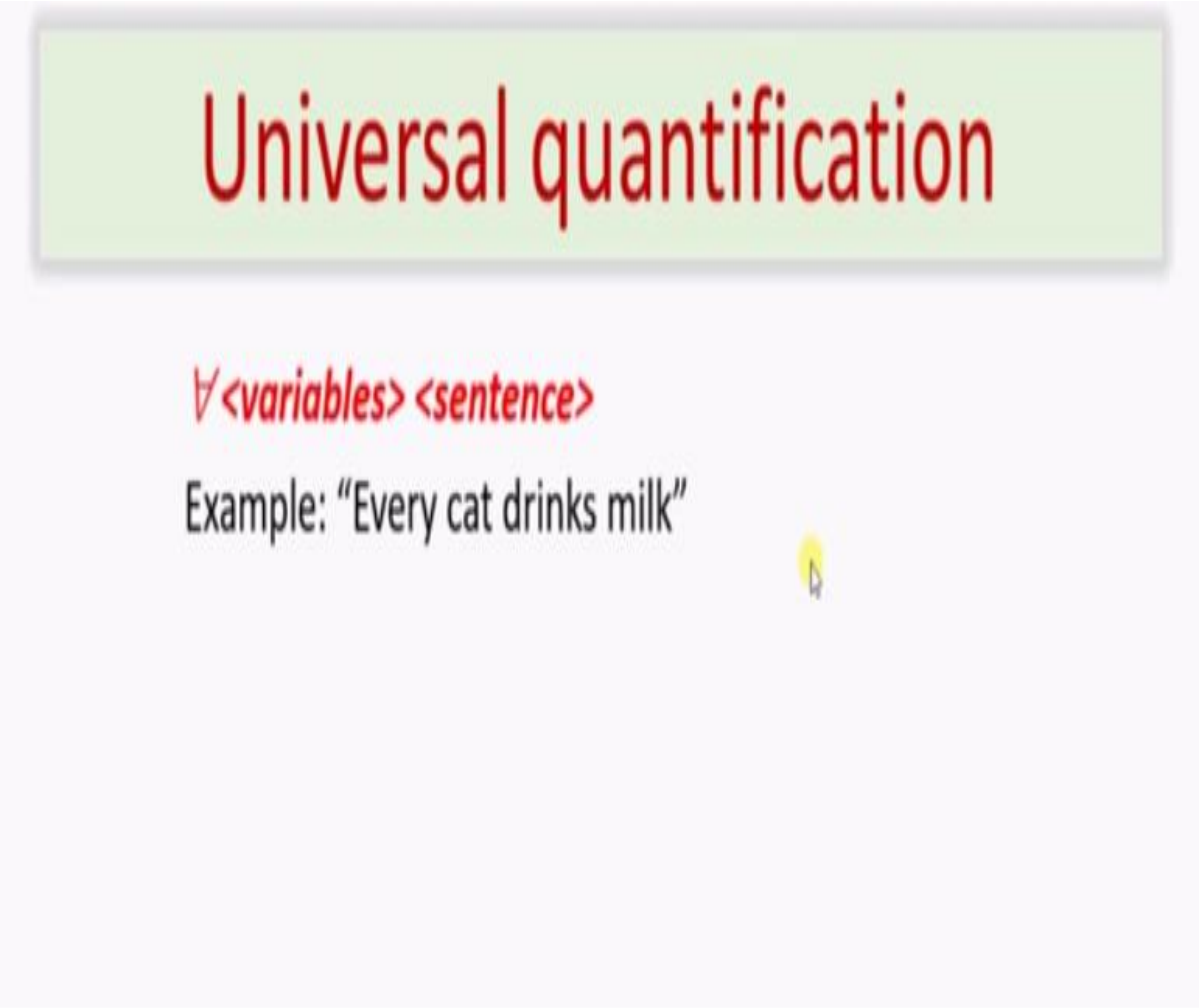
MOTHER(JOHN)

MARRIED(FATHER(RITA), MOTHER(JOHN))



Syntax of FOL: Basic elements

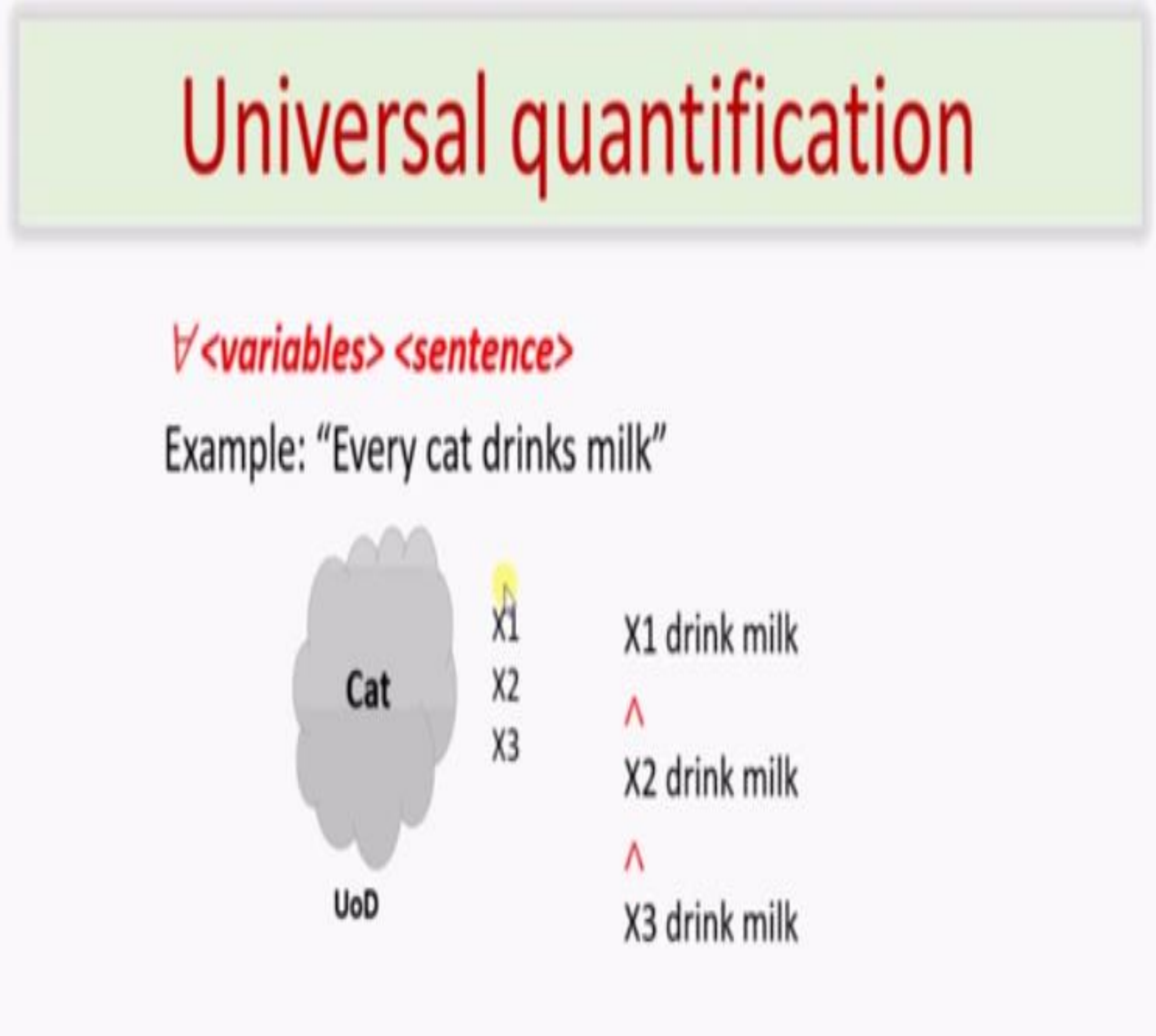
Constants	John, 2, Goa,...
Variables	x, y, a, b,...
Predicates	Brother, >,...
Function Relations	Sqrt, LeftLegOf,...
Connectives	¬, ⇒, ∧, ∨, ⇔
Equality	=
Quantifiers	∀, ∃



Universal quantification

∀<variables> <sentence>

Example: “Every cat drinks milk”



Universal quantification

∀ <variables> <sentence>

Example: "Every cat drinks milk"

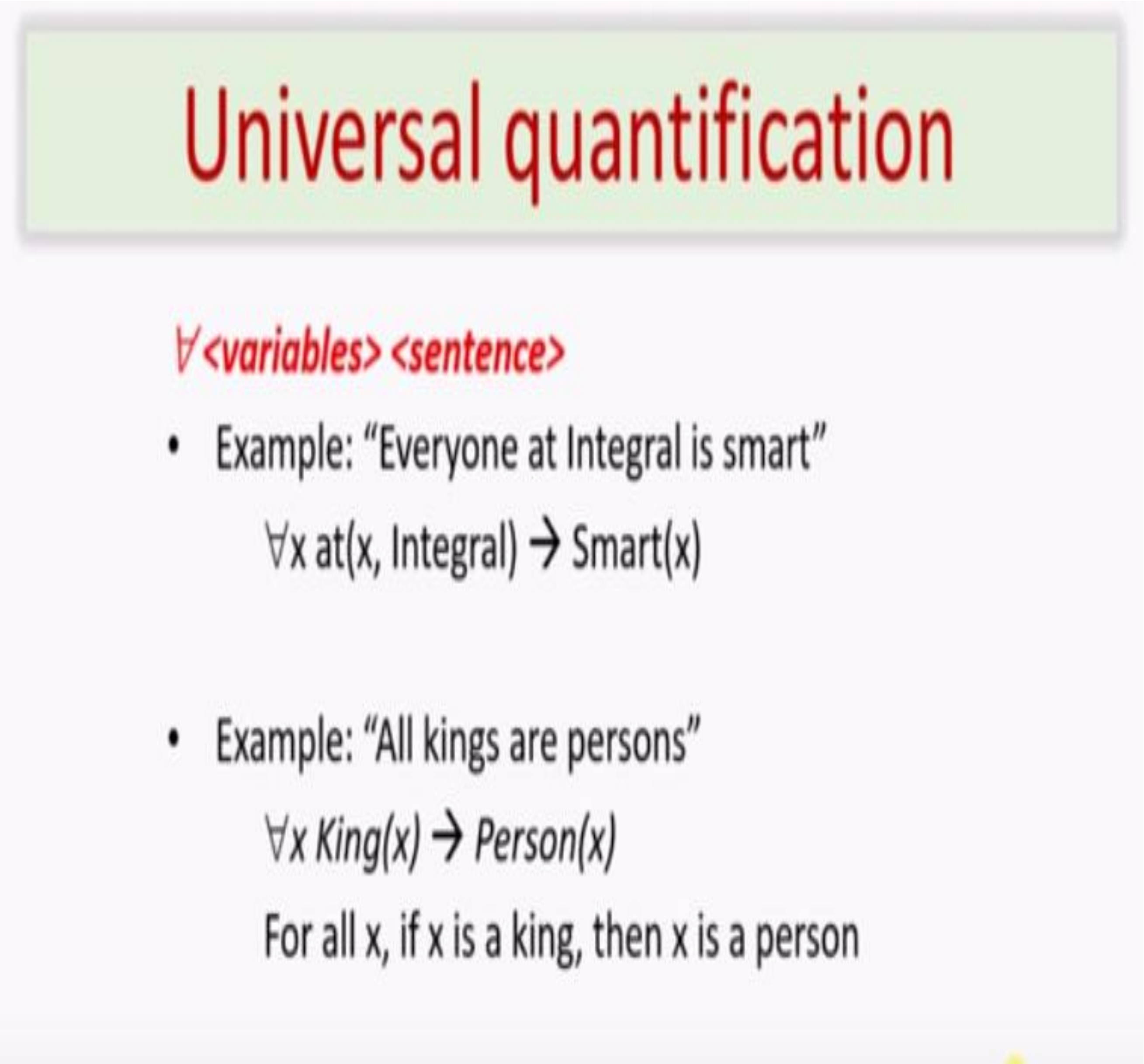
X1 drink milk

∧

X2 drink milk

∧

X3 drink milk



Universal quantification

$\forall$ *<variables> <sentence>*

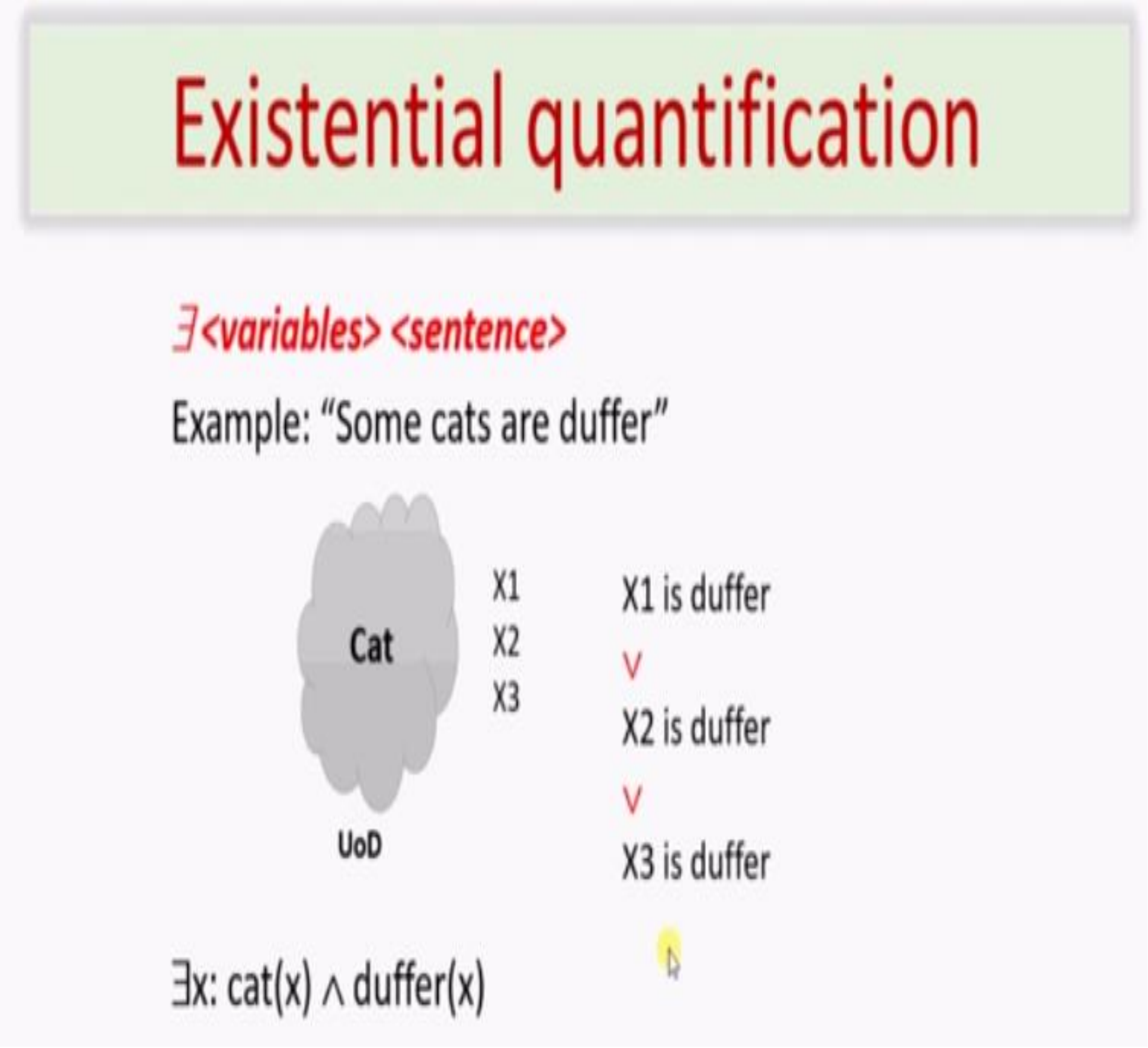
- Example: "Everyone at Integral is smart"

  $\forall x\ \mathrm{at}(x, \mathrm{Integral}) \rightarrow \mathrm{Smart}(x)$

- Example: "All kings are persons"

  $\forall x\ King(x) \rightarrow Person(x)$

  For all x, if x is a king, then x is a person



Existential quantification

∃ <variables> <sentence>

Example: "Some cats are duffer"

X1 is duffer

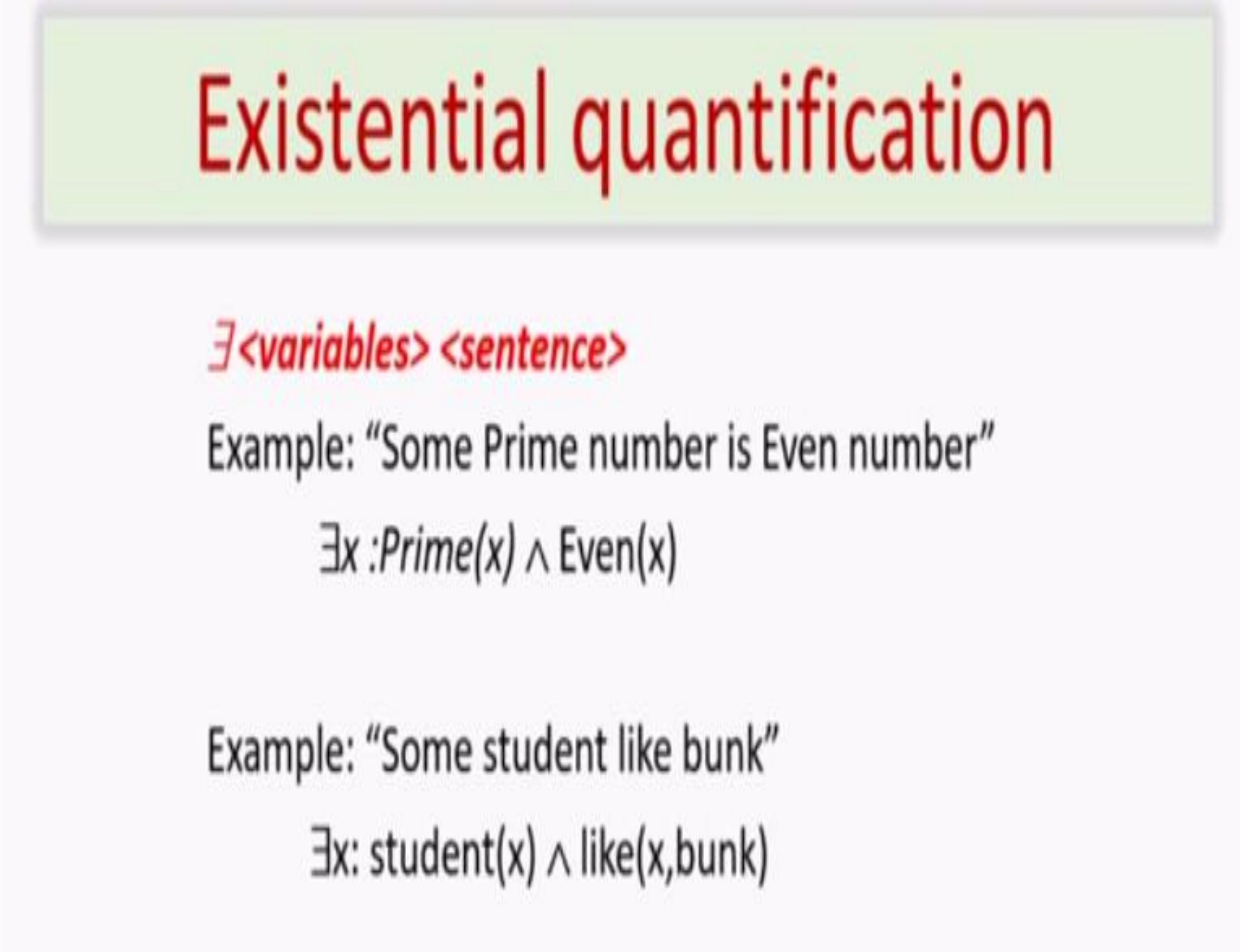
∨

X2 is duffer

∨

X3 is duffer

$\exists x$: cat(x) $\wedge$ duffer(x)



Existential quantification

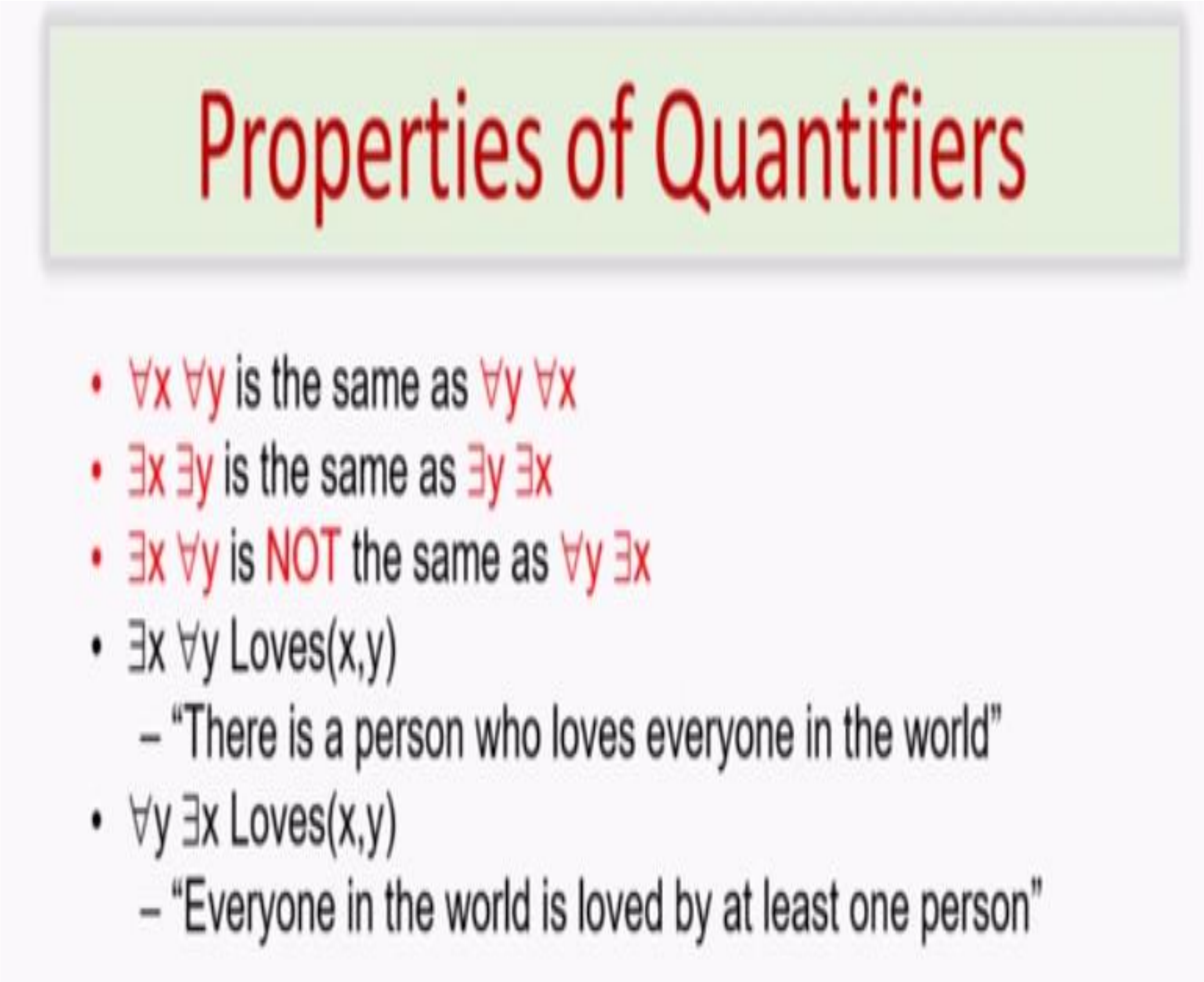
∃ <variables> <sentence>

Example: "Some Prime number is Even number"

$\exists x : \mathit{Prime(x)} \land \mathrm{Even}(x)$

Example: "Some student like bunk"

$\exists x: \mathrm{student}(x) \land \mathrm{like}(x,\mathrm{bunk})$

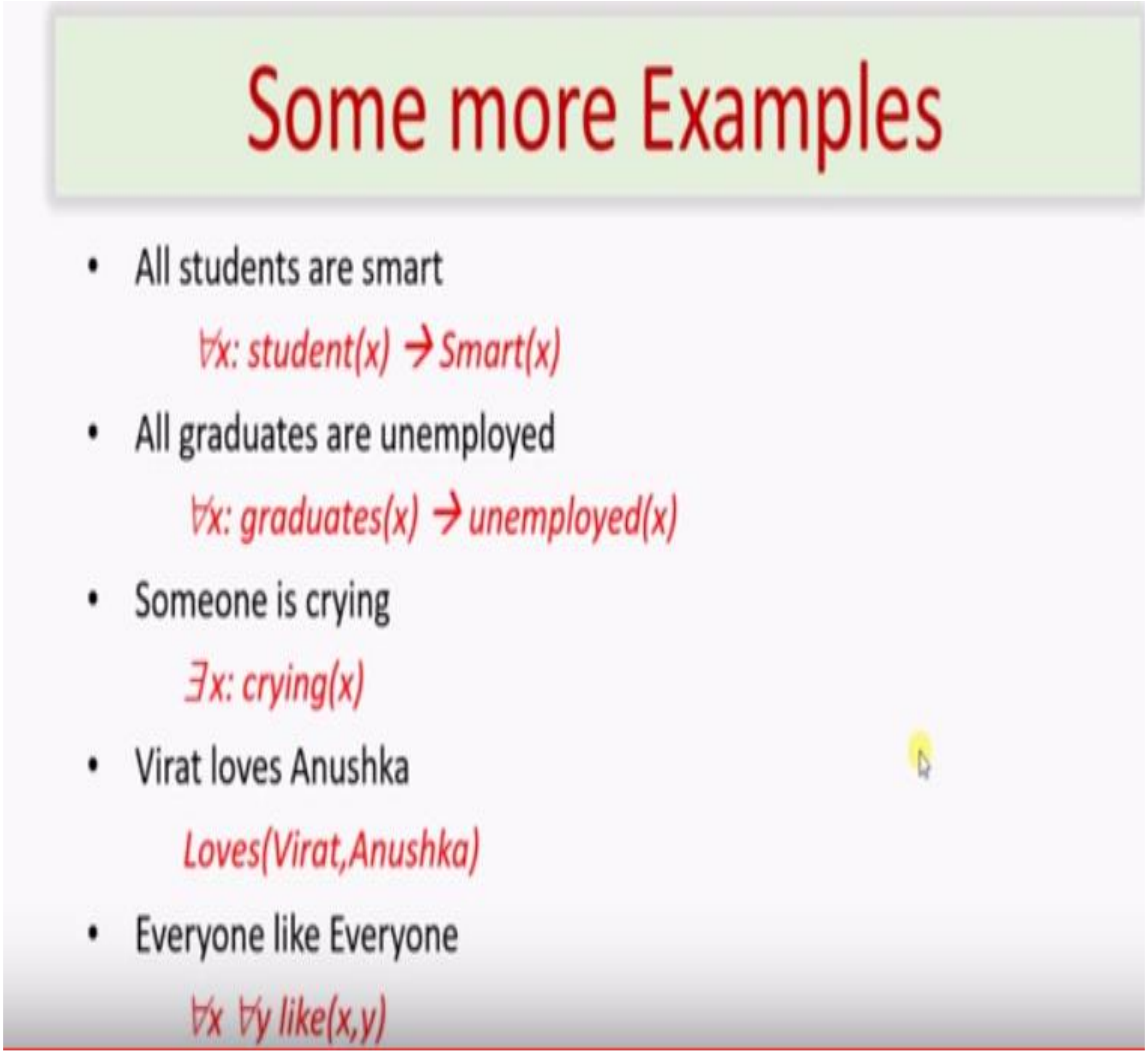


Properties of Quantifiers

- $\forall x\ \forall y$ is the same as $\forall y\ \forall x$
- $\exists x\ \exists y$ is the same as $\exists y\ \exists x$
- $\exists x\ \forall y$ is NOT the same as $\forall y\ \exists x$
- $\exists x\ \forall y$ Loves(x,y)
  - "There is a person who loves everyone in the world"
- $\forall y\ \exists x$ Loves(x,y)
  - "Everyone in the world is loved by at least one person"

FOL : EXERCISE TO SOLVE

- ALL STUDENTS ARE SMART
- ALL GRADUATES ARE UNEMPLOYED
- SOMEONE IS CRYING
- VIRAT LOVES ANUSHKA
- EVERYONE LIKE EVERYONE



Some more Examples

- All students are smart

  $\forall x: student(x) \rightarrow Smart(x)$

- All graduates are unemployed

  $\forall x: graduates(x) \rightarrow unemployed(x)$

- Someone is crying

  $\exists x: crying(x)$

- Virat loves Anushka

  $Loves(Virat,Anushka)$

- Everyone like Everyone

  $\forall x \ \forall y \ like(x,y)$

ENTAILMENT

If X, then Y.

Jim rides a bike to school every morning.
Jim is good at riding bikes.

ENTAILMENT

- **GIVEN 2 SENTENCES P AND Q WE SAY Q ENTAILS Q** **,** $P \mid = Q$ IF Q HOLDS IN EVERY MODEL THAT P HOLDS.
- **ONE THING FOLLOWS FROM ANOTHER**

$$KB \mid = A$$

- KB ENTAILS SENTENCE A ***IF AND ONLY IF*** **A IS TRUE** IN **WORLDS WHERE KB IS TRUE.**
- E.G. X+Y=4 ENTAILS 4=X+Y
- **ENTAILMENT** IS A **RELATIONSHIP BETWEEN SENTENCES THAT IS BASED ON SEMANTICS**.

Validity and satisfiability

A sentence is valid if it is true in **all** models,

e.g., *True*, $A \lor \neg A$, $A \Rightarrow A$, $(A \land (A \Rightarrow B)) \Rightarrow B$

Validity is connected to inference via the Deduction Theorem:

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

A sentence is satisfiable if it is true in **some** model

e.g., $A \lor B$, C

A sentence is unsatisfiable if it is true in **no** models

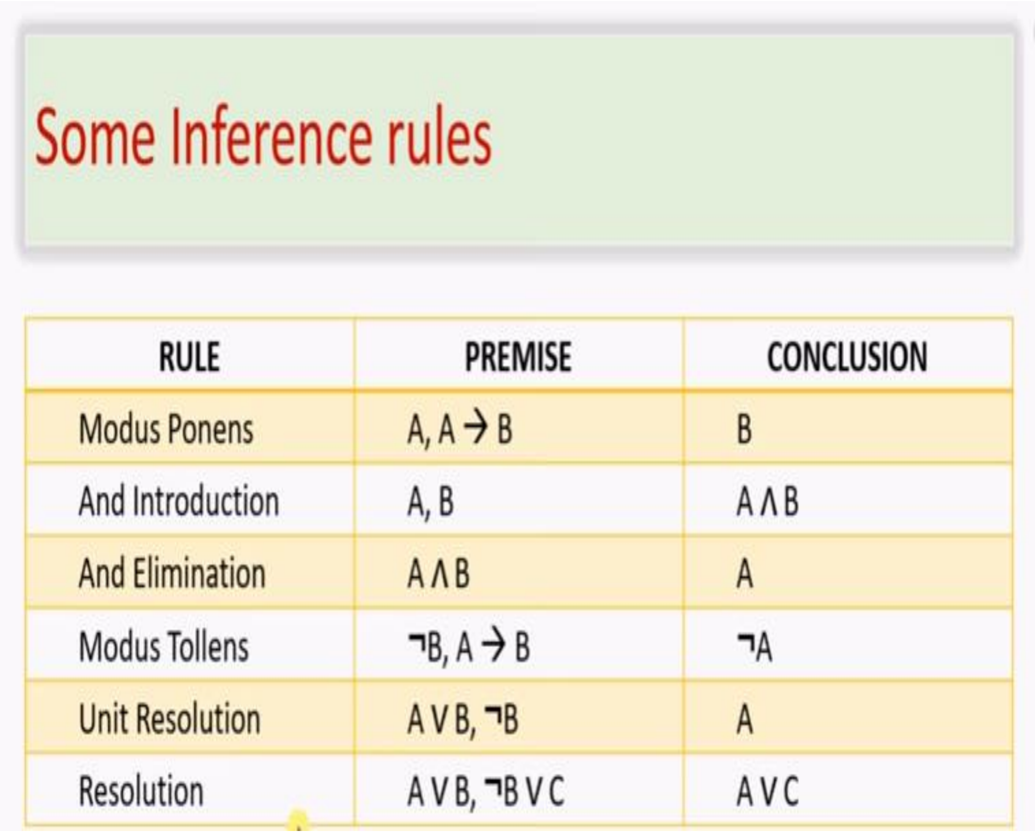
e.g., $A \land \neg A$

Satisfiability is connected to inference via the following:

$KB \models \alpha$ if and only if $(KB \land \neg \alpha)$ is unsatisfiable

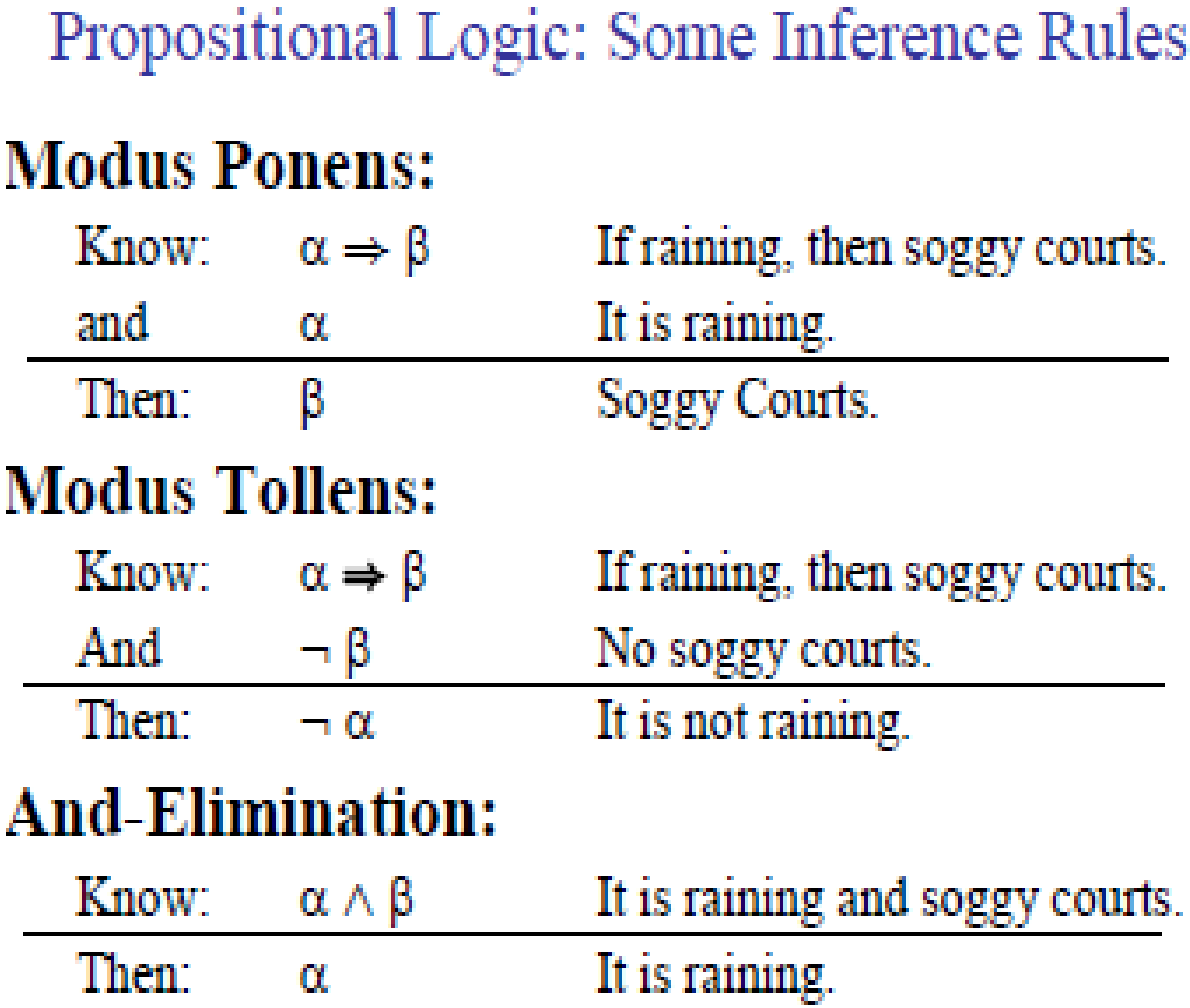
Sound rules of inference

- Here are some examples of sound rules of inference
  - *A rule is sound if its conclusion is true*



Some Inference rules

RULE	PREMISE	CONCLUSION
Modus Ponens	$A, A \rightarrow B$	B
And Introduction	A, B	$A \wedge B$
And Elimination	$A \wedge B$	A
Modus Tollens	$\neg B, A \rightarrow B$	$\neg A$
Unit Resolution	$A \vee B, \neg B$	A
Resolution	$A \vee B, \neg B \vee C$	$A \vee C$



Propositional Logic: Some Inference Rules

Modus Ponens:

Know:	$\alpha \Rightarrow \beta$	If raining, then soggy courts.
and	α	It is raining.
Then:	β	Soggy Courts.

Modus Tollens:

Know:	$\alpha \Rightarrow \beta$	If raining, then soggy courts.
And	$\neg \beta$	No soggy courts.
Then:	$\neg \alpha$	It is not raining.

And-Elimination:

Know:	$\alpha \wedge \beta$	It is raining and soggy courts.
Then:	α	It is raining.

Resolution

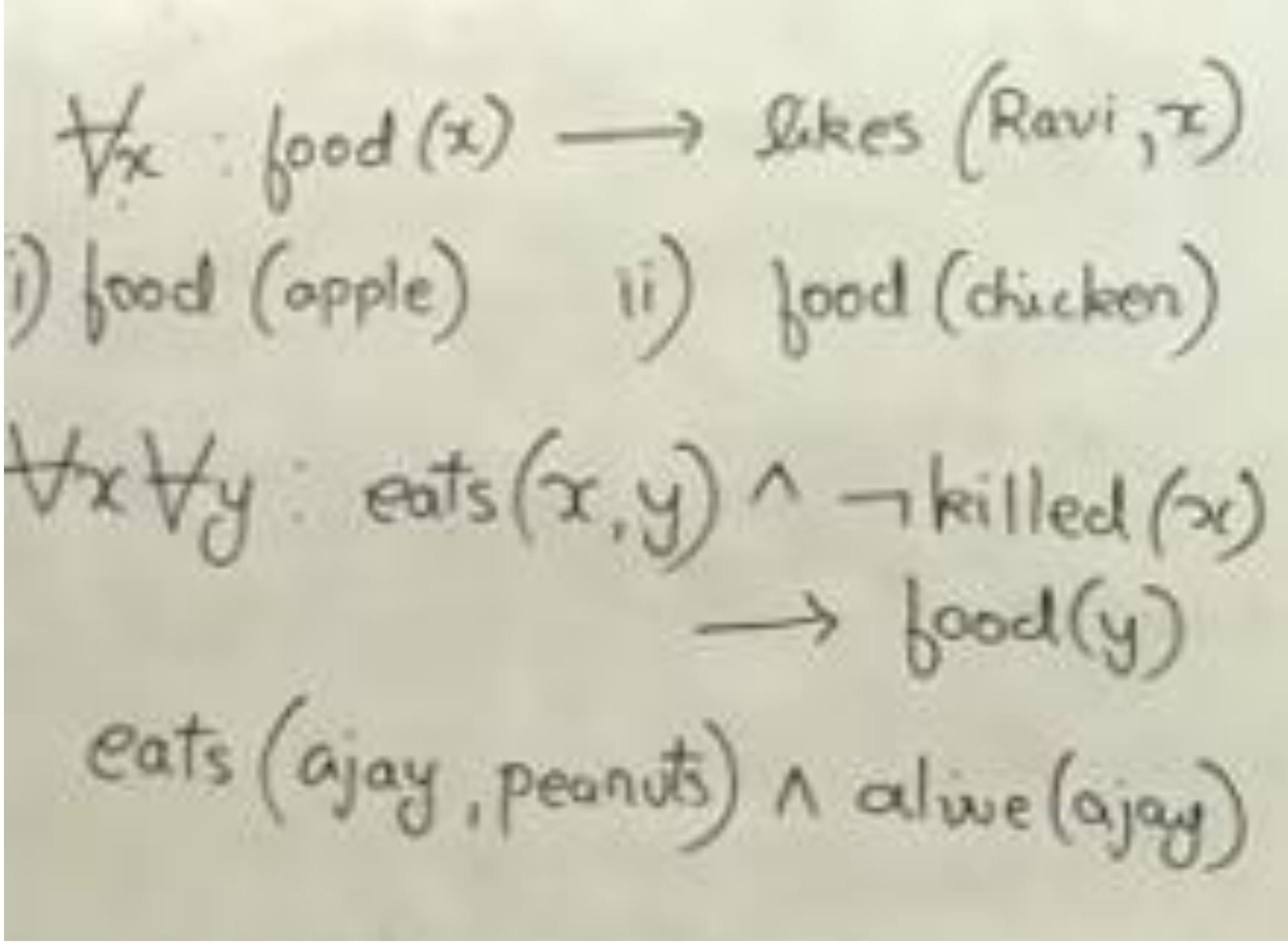
1. Ravi likes all kind of food
2. Apple and chicken are food
3. Anything anyone eats and is not killed is a food
4. Ajay eats peanut and still alive
5. Rita eats that Ajay eats

Prove : Ravi likes peanuts likes(Ravi , Peanut)

Resolution steps

- Negate the statement to be proved
- Convert facts into FOL
- Convert FOL into CNF
- Draw resolution graph

Convert to FOL



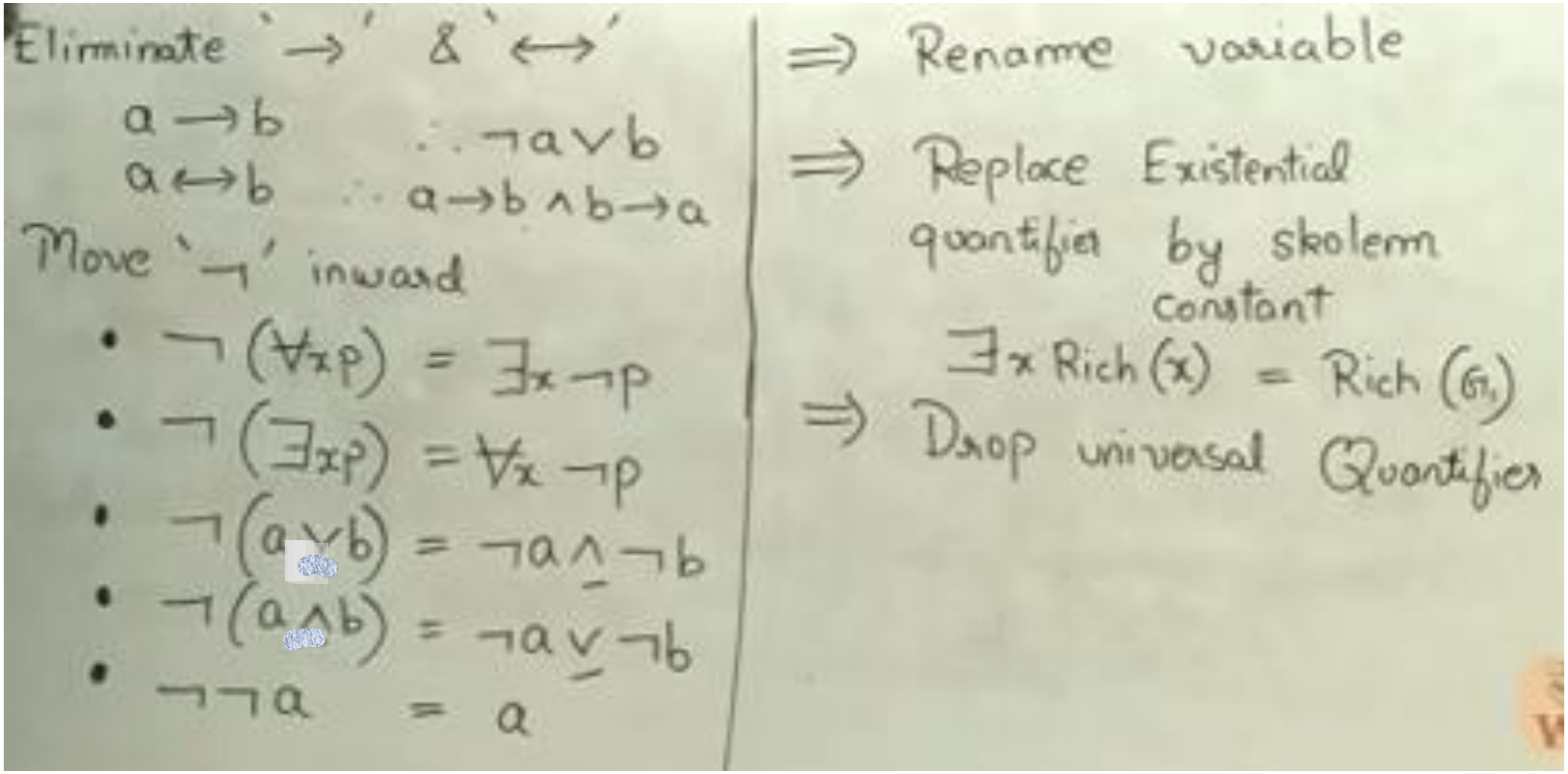
$\forall x : food(x) \longrightarrow likes(Ravi, x)$

i) $food(apple)$ ii) $food(chicken)$

$\forall x \forall y : eats(x, y) \wedge \neg killed(x) \longrightarrow food(y)$

$eats(ajay, peanuts) \wedge alive(ajay)$

Rules to convert FOL in CNF



Eliminate '→' & '↔'

$a \rightarrow b \quad \therefore \neg a \vee b$

$a \leftrightarrow b \quad \therefore a \rightarrow b \wedge b \rightarrow a$

Move '¬' inward

- $\neg(\forall x p) = \exists x \neg p$
- $\neg(\exists x p) = \forall x \neg p$
- $\neg(a \vee b) = \neg a \underline{\wedge} \neg b$
- $\neg(a \wedge b) = \neg a \underline{\vee} \neg b$
- $\neg\neg a = a$

⇒ Rename variable

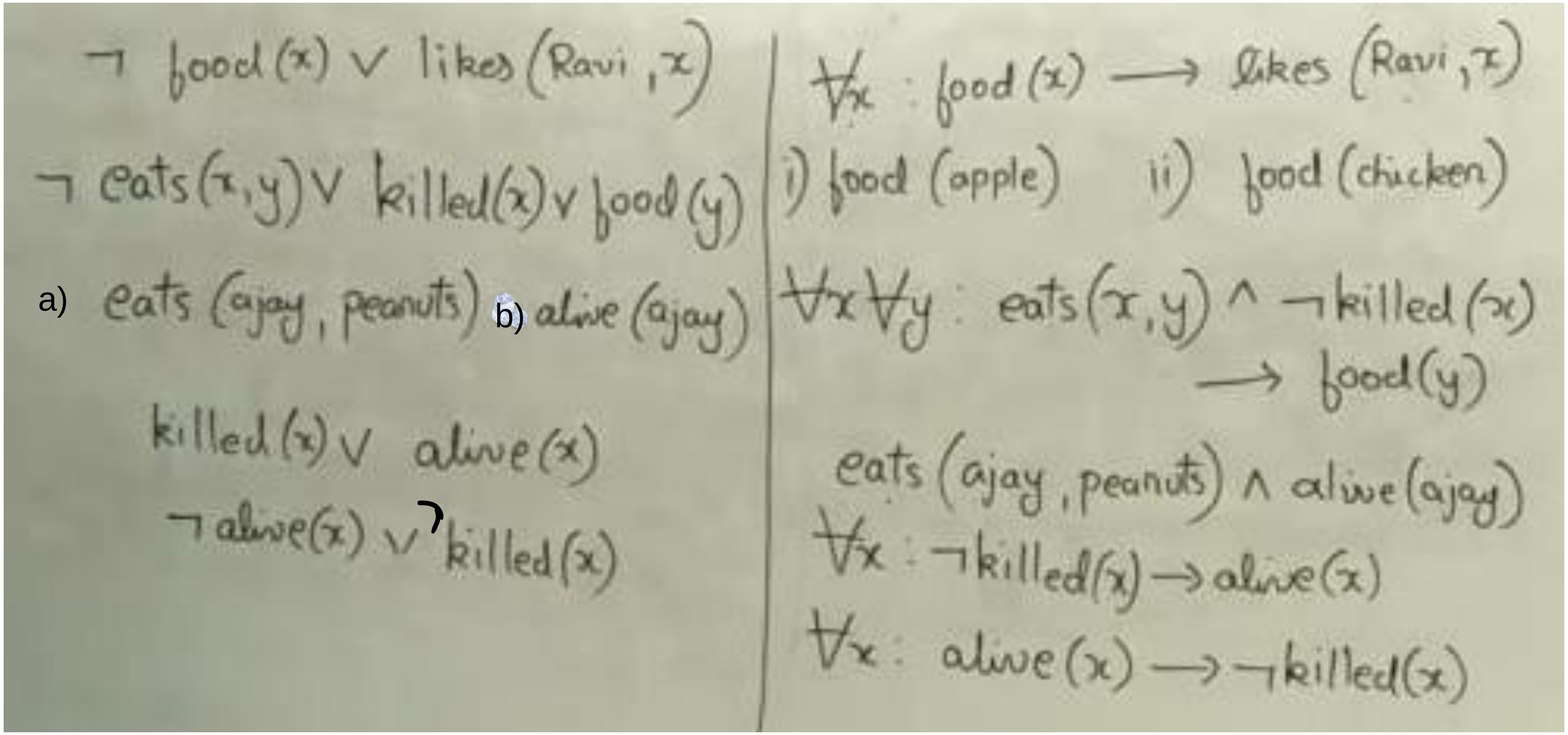
⇒ Replace Existential quantifier by skolem constant

$\exists x \, Rich(x) = Rich(G_1)$

⇒ Drop universal Quantifier

In CNF no “and (^) ” operator is available
If ^ operator is there make it as new statement

Convert FOL to CNF



$\neg\, food(x) \vee likes(Ravi, x)$

$\neg\, eats(x,y) \vee killed(x) \vee food(y)$

a) $eats(ajay, peanuts)$ b) $alive(ajay)$

$killed(x) \vee alive(x)$

$\neg alive(x) \vee \neg killed(x)$

$\forall x : food(x) \rightarrow likes(Ravi, x)$

i) $food(apple)$ ii) $food(chicken)$

$\forall x \forall y : eats(x,y) \wedge \neg killed(x) \rightarrow food(y)$

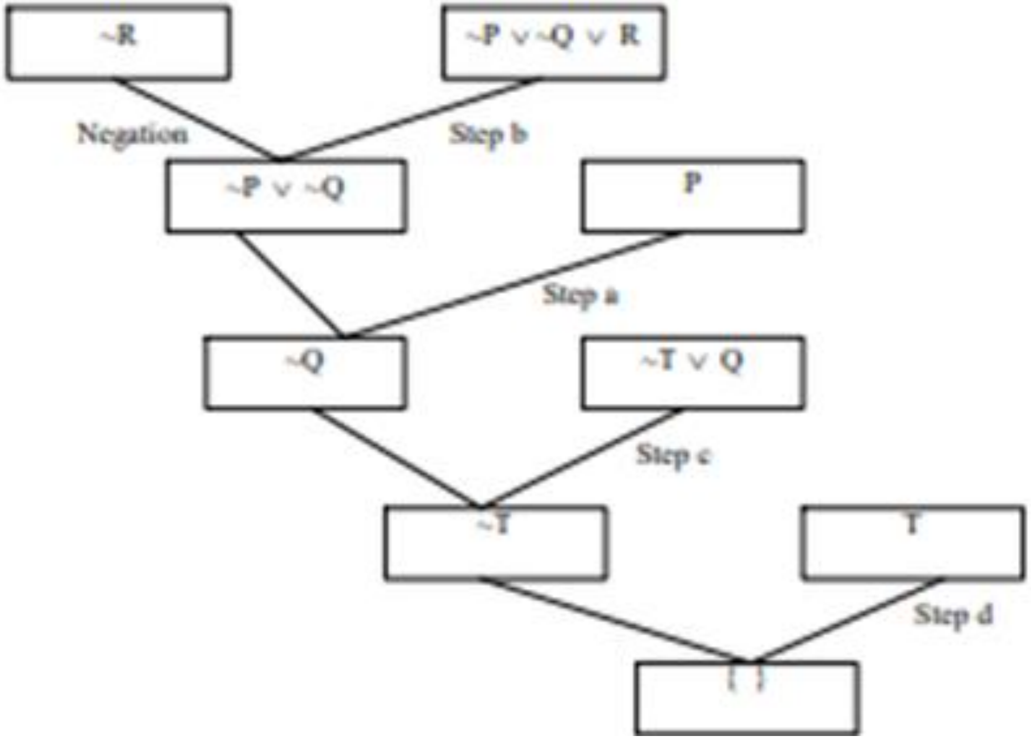
$eats(ajay, peanuts) \wedge alive(ajay)$

$\forall x : \neg killed(x) \rightarrow alive(x)$

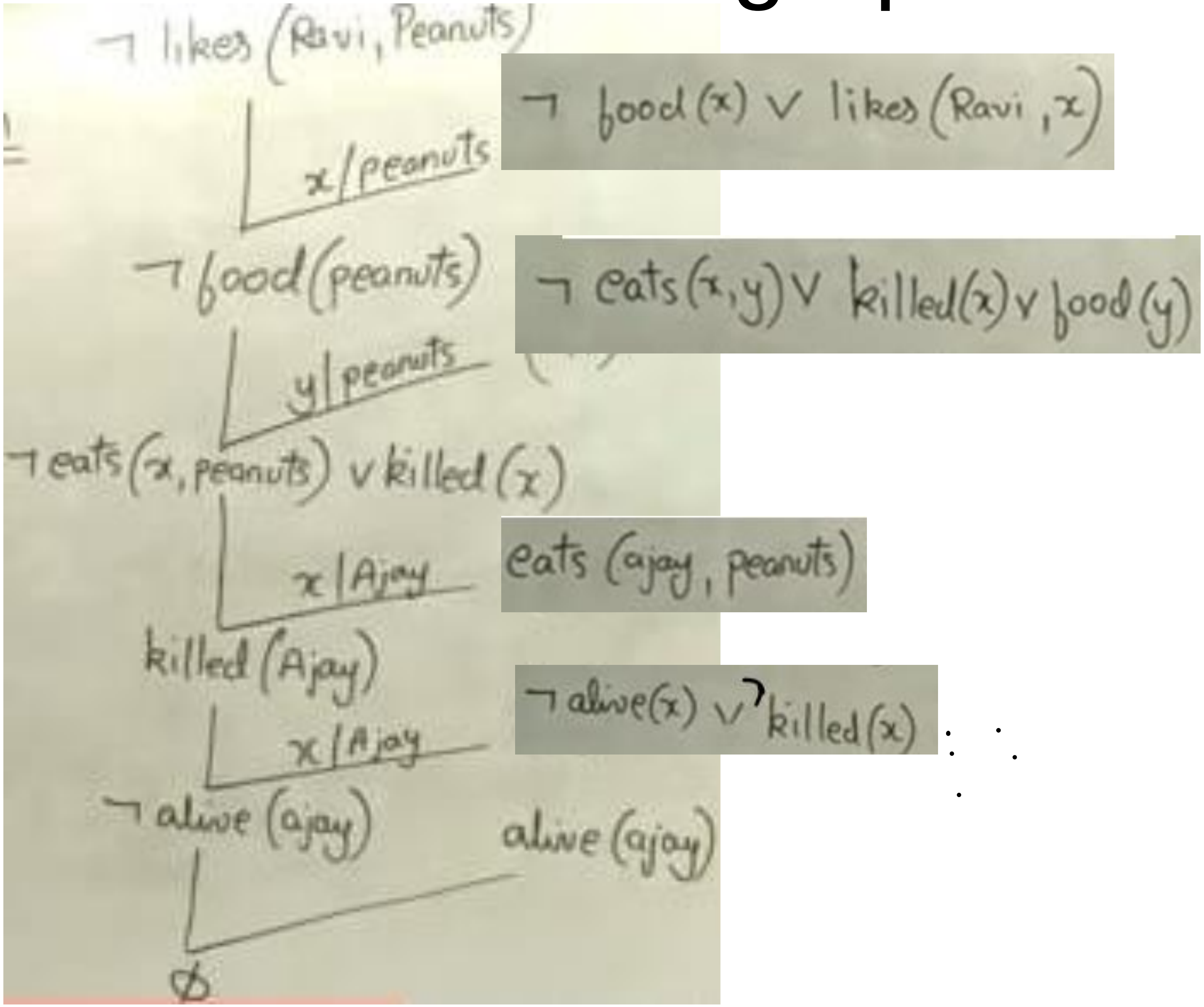
$\forall x : alive(x) \rightarrow \neg killed(x)$

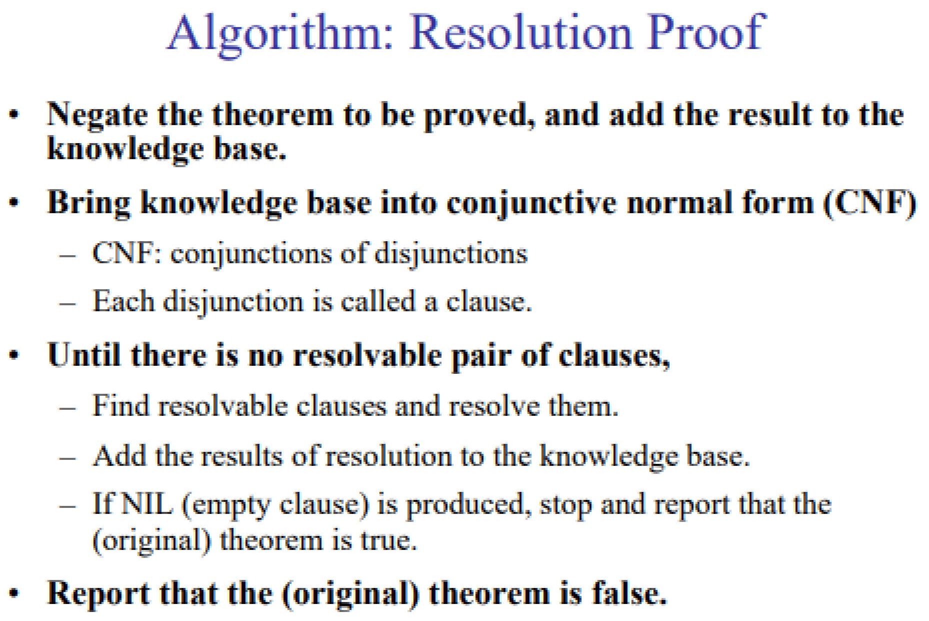
In CNF no “and (^) ” operator is available
If ^ operator is there make it as new statement
(check in third statement)

Example : Resolution Graph

- 

Draw resolution graph





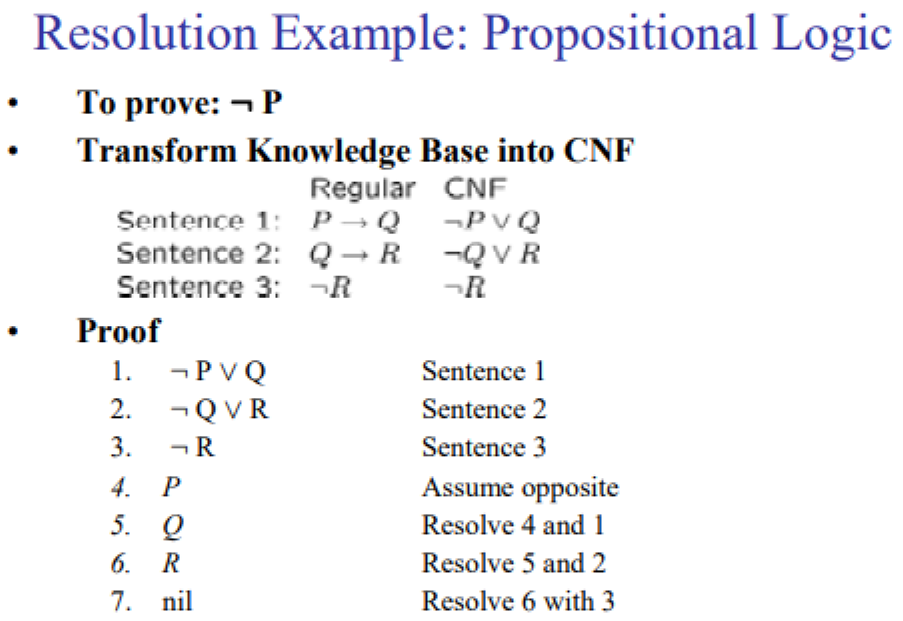
Algorithm: Resolution Proof

- **Negate the theorem to be proved, and add the result to the knowledge base.**
- **Bring knowledge base into conjunctive normal form (CNF)**
  - CNF: conjunctions of disjunctions
  - Each disjunction is called a clause.
- **Until there is no resolvable pair of clauses,**
  - Find resolvable clauses and resolve them.
  - Add the results of resolution to the knowledge base.
  - If NIL (empty clause) is produced, stop and report that the (original) theorem is true.
- **Report that the (original) theorem is false.**

Methods used for resolution technique

- **Resolution Proof:-** A set of inference rules and resolution rules can be used to derive a conclusion from the KB.
- **Resolution with refutation (or) proof by contradiction (or) reduction and absurdum:-** One complete inference procedure using resolution is refutation. The idea is to prove P, we assume P is false (i.e. add ¬P to KB) and prove a contradiction, that is the KB implies P.

$$\mathbf{(KB \wedge \neg P \Rightarrow False) \Leftrightarrow (KB \Rightarrow P)}$$



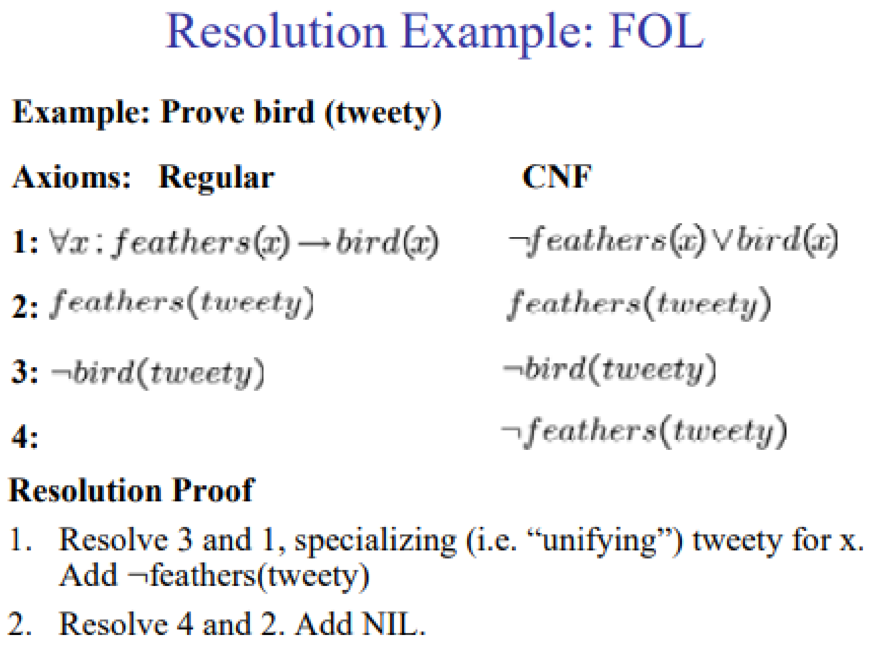
Resolution Example: Propositional Logic

- **To prove: ¬ P**
- **Transform Knowledge Base into CNF**

	Regular	CNF
Sentence 1:	$P \rightarrow Q$	$\neg P \vee Q$
Sentence 2:	$Q \rightarrow R$	$\neg Q \vee R$
Sentence 3:	$\neg R$	$\neg R$

- **Proof**

1.	¬ P ∨ Q	Sentence 1
2.	¬ Q ∨ R	Sentence 2
3.	¬ R	Sentence 3
4.	*P*	Assume opposite
5.	*Q*	Resolve 4 and 1
6.	*R*	Resolve 5 and 2
7.	nil	Resolve 6 with 3



Resolution Example: FOL

Example: Prove bird (tweety)

Axioms:	**Regular**	**CNF**
1:	$\forall x : feathers(x) \rightarrow bird(x)$	$\neg feathers(x) \vee bird(x)$
2:	$feathers(tweety)$	$feathers(tweety)$
3:	$\neg bird(tweety)$	$\neg bird(tweety)$
4:		$\neg feathers(tweety)$

Resolution Proof

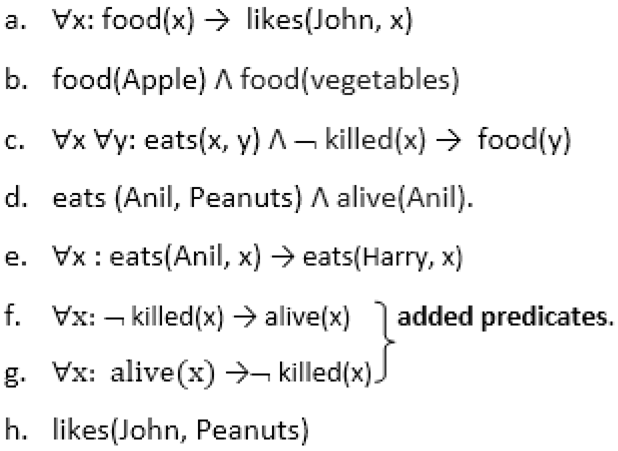
1. Resolve 3 and 1, specializing (i.e. "unifying") tweety for x. Add ¬feathers(tweety)
2. Resolve 4 and 2. Add NIL.

Example 2

1.John likes all kind of food.
2.Apple and vegetable are food
3.Anything anyone eats and not killed is food.
4.Anil eats peanuts and still alive
5.Harry eats everything that Anil eats.

Prove by resolution that:
1.John likes peanuts.

FOL



a. ∀x: food(x) → likes(John, x)

b. food(Apple) Λ food(vegetables)

c. ∀x ∀y: eats(x, y) Λ ¬ killed(x) → food(y)

d. eats (Anil, Peanuts) Λ alive(Anil).

e. ∀x : eats(Anil, x) → eats(Harry, x)

f. ∀x: ¬ killed(x) → alive(x)

g. ∀x: alive(x) →¬ killed(x)

added predicates.

h. likes(John, Peanuts)

FOL to CNF

a. ¬ food(x) V likes(John, x)

b. food(Apple)

c. food(vegetables)

d. ¬ eats(y, z) V killed(y) V food(z)

e. eats (Anil, Peanuts)

f. alive(Anil)

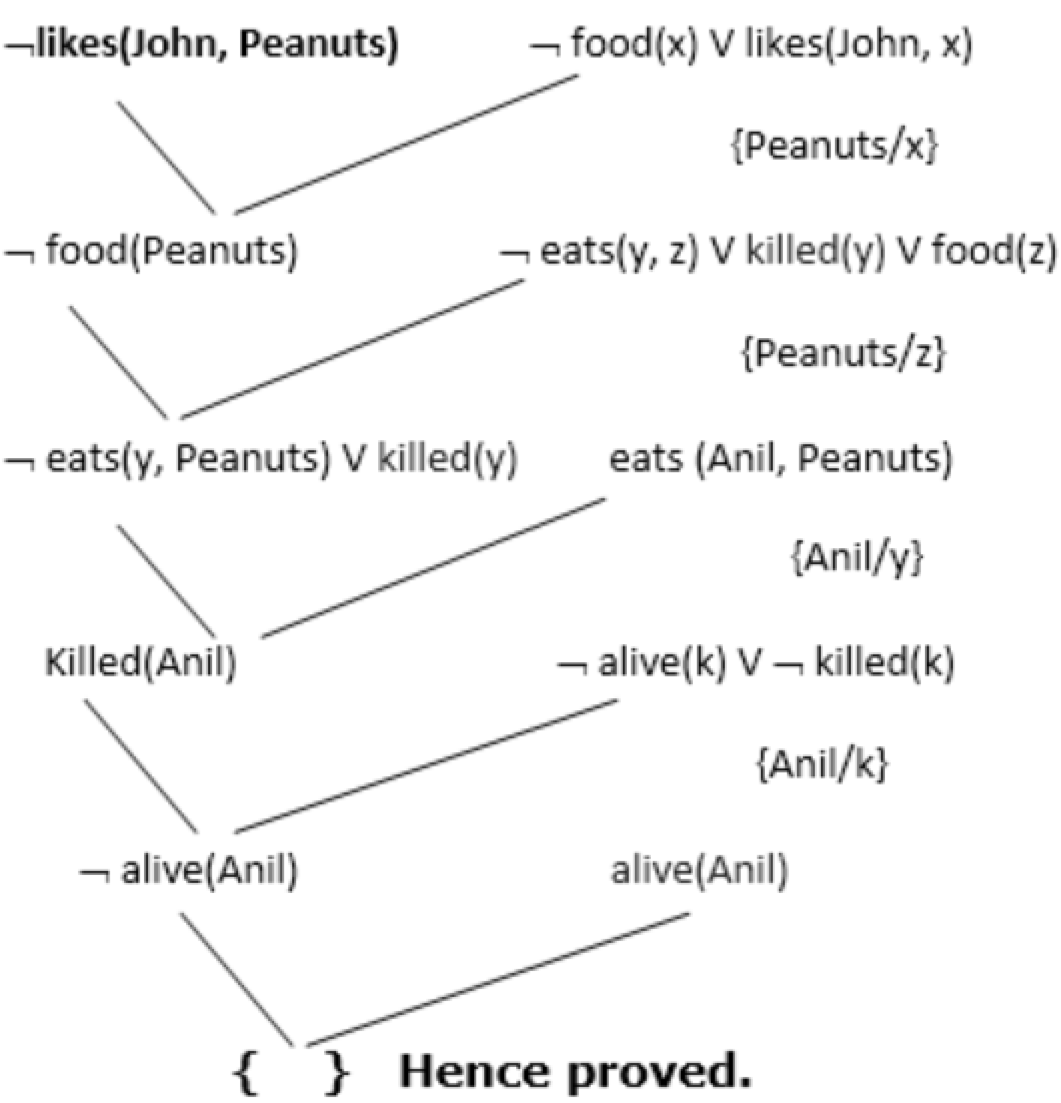
g. ¬ eats(Anil, w) V eats(Harry, w)

h. killed(g) V alive(g)

i. ¬ alive(k) V ¬ killed(k)

j. likes(John, Peanuts).

Resolution Graph



¬likes(John, Peanuts)

¬ food(x) V likes(John, x)

{Peanuts/x}

¬ food(Peanuts)

¬ eats(y, z) V killed(y) V food(z)

{Peanuts/z}

¬ eats(y, Peanuts) V killed(y)

eats (Anil, Peanuts)

{Anil/y}

Killed(Anil)

¬ alive(k) V ¬ killed(k)

{Anil/k}

¬ alive(Anil)

alive(Anil)

{ } Hence proved.

Exercises : Resolution

- Convert the given facts to predicate logic and prove that the statement : "Ram did not jump" is false using Resolution Graph.
  - Ram went to temple
  - The way to temple is, walk till post box and take left or right road.
  - The left road has a ditch
  - Way to cross the ditch is to jump
  - A log is across the right road
  - One needs to jump across the log to go ahead.

Exercises : Resolution

- Convert the given facts to predicate logic and CNF. Also, prove that the statement : “Raja is angry” using Resolution Graph.
  - Rimi is hungry
  - If Rimi is hungry she barks
  - If Rimi is barking then Raja is angry

Exercises : Resolution

- Convert the given facts to FOL and CNF. Also, prove that the statement : "Someone is Smiling" using Resolution technique and draw the resolution tree.
  - All people who are graduating are happy
  - All happy people smile
  - Someone is graduating.

Inference Engine

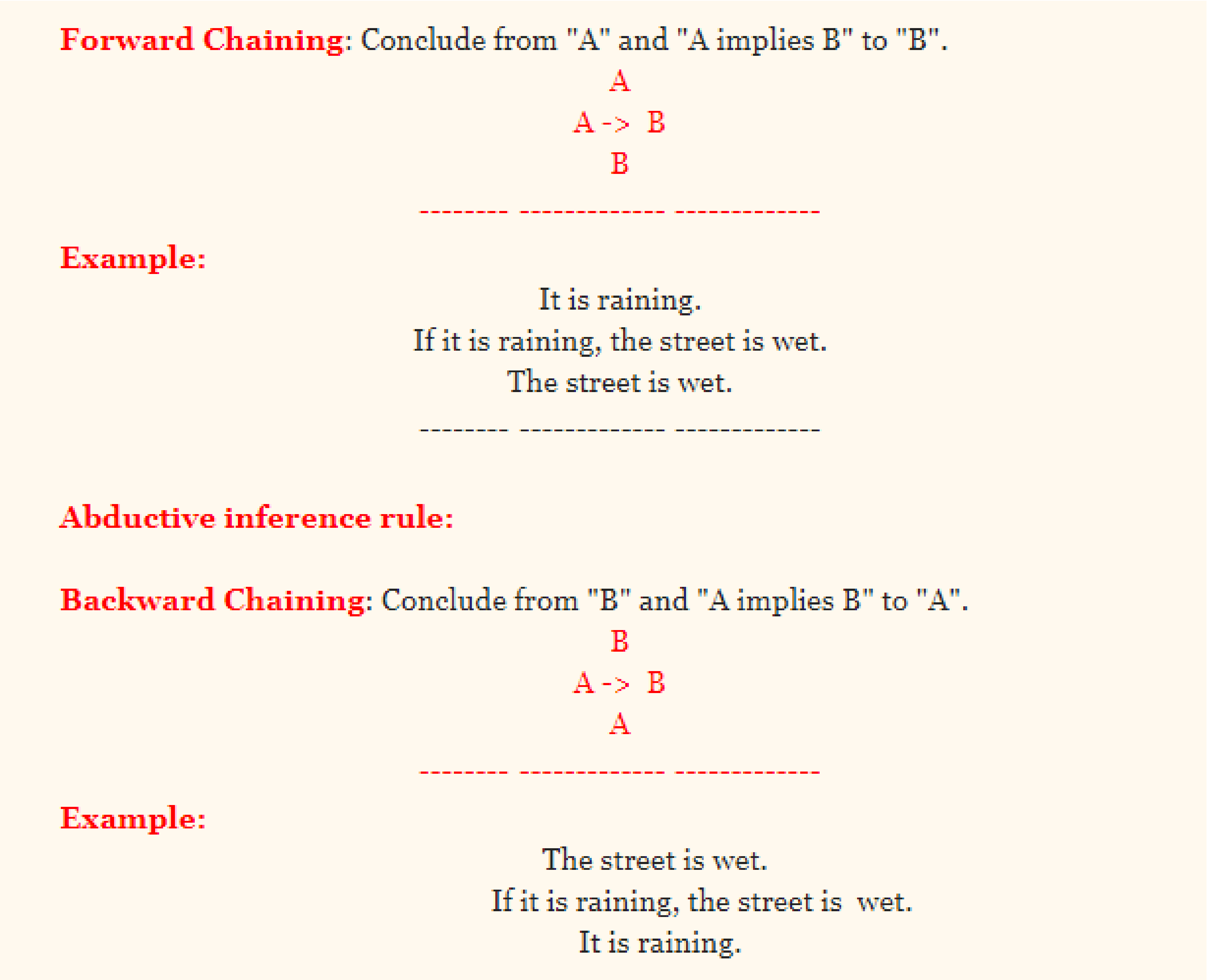
- **Core component** of the intelligent system
- Applies logical rules to the knowledge base to infer new information from known facts.
- The first inference engine was part of the expert system.
- Inference engine commonly proceeds in two modes :
  - **Forward chaining**
  - **Backward chaining**

Forward Chaining / Forward Deduction / Forward Reasoning method

- Starts with known facts(if part)
- Asserts new facts

Eg. : If it is raining then we need umbrella.

- It is raining - data
- We need umbrella – decision (new facts)



Forward Chaining: Conclude from "A" and "A implies B" to "B".

A

A -> B

B

-------- ------------ ------------

Example:

It is raining.

If it is raining, the street is wet.

The street is wet.

-------- ------------ ------------

Abductive inference rule:

Backward Chaining: Conclude from "B" and "A implies B" to "A".

B

A -> B

A

-------- ------------ ------------

Example:

The street is wet.

If it is raining, the street is wet.

It is raining.

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.
- Also called as data-driven as we reach to the goal using available data.
- Commonly used in the expert system, such as CLIPS, business, and production rule systems.

Example of forward chaining

- Example: KB = All cats like fish, cats eat everything they like, and Ziggy is a cat. In FOL, KB =

```
1.  (Ax) cat(x) => likes(x, Fish)
2.  (Ax)(Ay) (cat(x) ^ likes(x,y)) => eats(x,y)
3.  cat(Ziggy)
```

- Goal query: Does Ziggy eat fish?

Generalized Modus Ponens (GMP)

$$\frac{p_1', p_2', \ldots, p_n', (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{\text{Subst}(\theta, q)}$$

where we can unify p_i‘ and p_i for all *i*

Example:

$$\frac{\mathit{King}(\mathit{John}), \mathit{Greedy}(\mathit{John}) \quad , \forall x\ \text{King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)}{\mathit{Evil}(\mathit{John})}$$

p_1' is $\mathit{King}(\mathit{John})$ $\quad$ p_1 is $\mathit{King}(x)$

p_2' is $\mathit{Greedy}(\mathit{John})$ $\quad$ p_2 is $\mathit{Greedy}(x)$

θ is $\{x/\text{John}\}$ $\quad$ q is $\mathit{Evil}(x)$

$\text{Subst}(\theta, q)$ is $\mathit{Evil}(\mathit{John})$

Proof: Data-driven

1. Use GMP with (1) and (3) to derive: 4. `likes(Ziggy, Fish)`
2. Use GMP with (3), (4) and (2) to derive `eats(Ziggy, Fish)`
3. So, Yes, Ziggy eats fish.

Example : Forward Chaining

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

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

Eg. : Facts Conversion into FOL

- It is a crime for an American to sell weapons to hostile nations. (Let's say p, q, and r are variables)
  $American(p) \wedge weapon(q) \wedge sells(p, q, r) \wedge hostile(r) \rightarrow Criminal(p)$ ...(1)
- CountryA owns some missiles.

  $\exists m\ Owns(A, m) \wedge Missile(m)$.
- It can be written in two definite clauses by using Existential Instantiation, introducing new Constant T1.
  $Owns(A, T1)$ ......(2)
  $Missile(T1)$ ....(3)

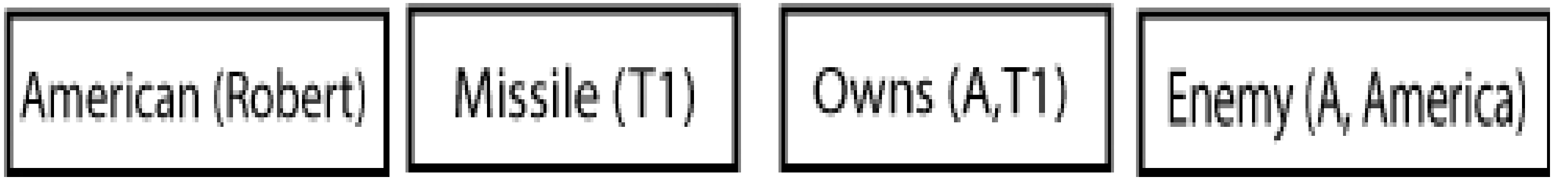
Eg. : Facts Conversion into FOL

- All of the missiles were sold to countryA by Robert.
  ∃m Missiles(m) ∧ Owns (A, m) → Sells (Robert, m,A) .....(4)
- Missiles are weapons.
  Missile(m) → Weapons (m) ....(5)
- Enemy of America is known as hostile.
  Enemy(p, America) →Hostile(p) ....(6)
- Country A is an enemy of America.
  Enemy (A, America) ....(7)
- Robert is American
  American(Robert). ....(8)

Forward chaining proof:

- **Step-1:**

Don’t have implication



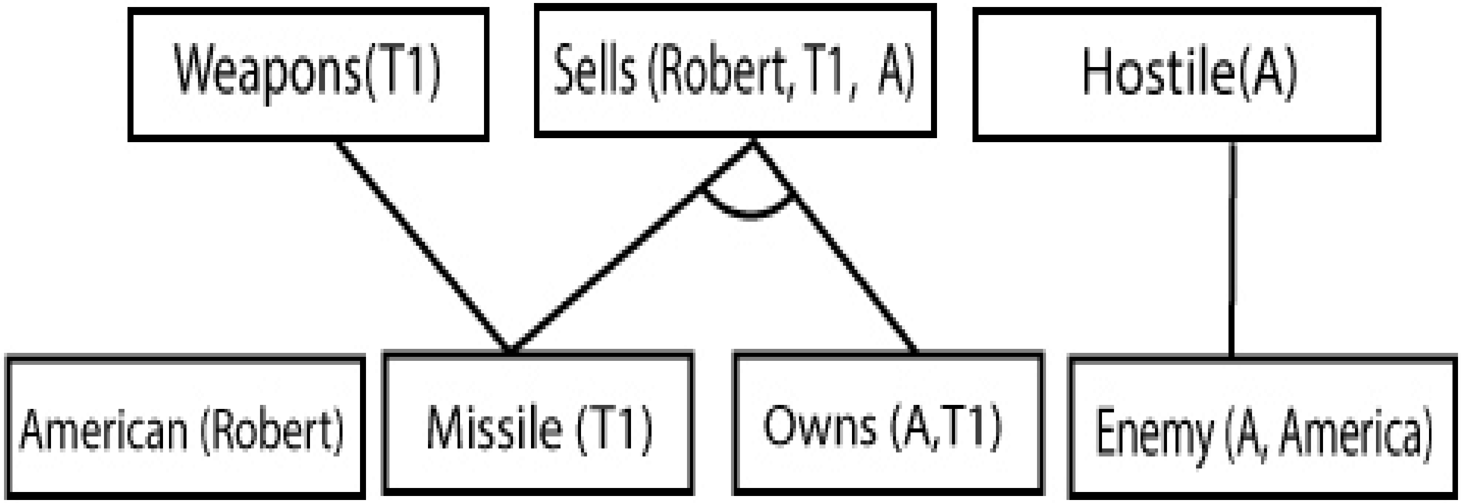
American (Robert)

Missile (T1)

Owns (A,T1)

Enemy (A, America)

- **Step-2:**



Weapons(T1)

Sells (Robert, T1, A)

Hostile(A)

American (Robert)

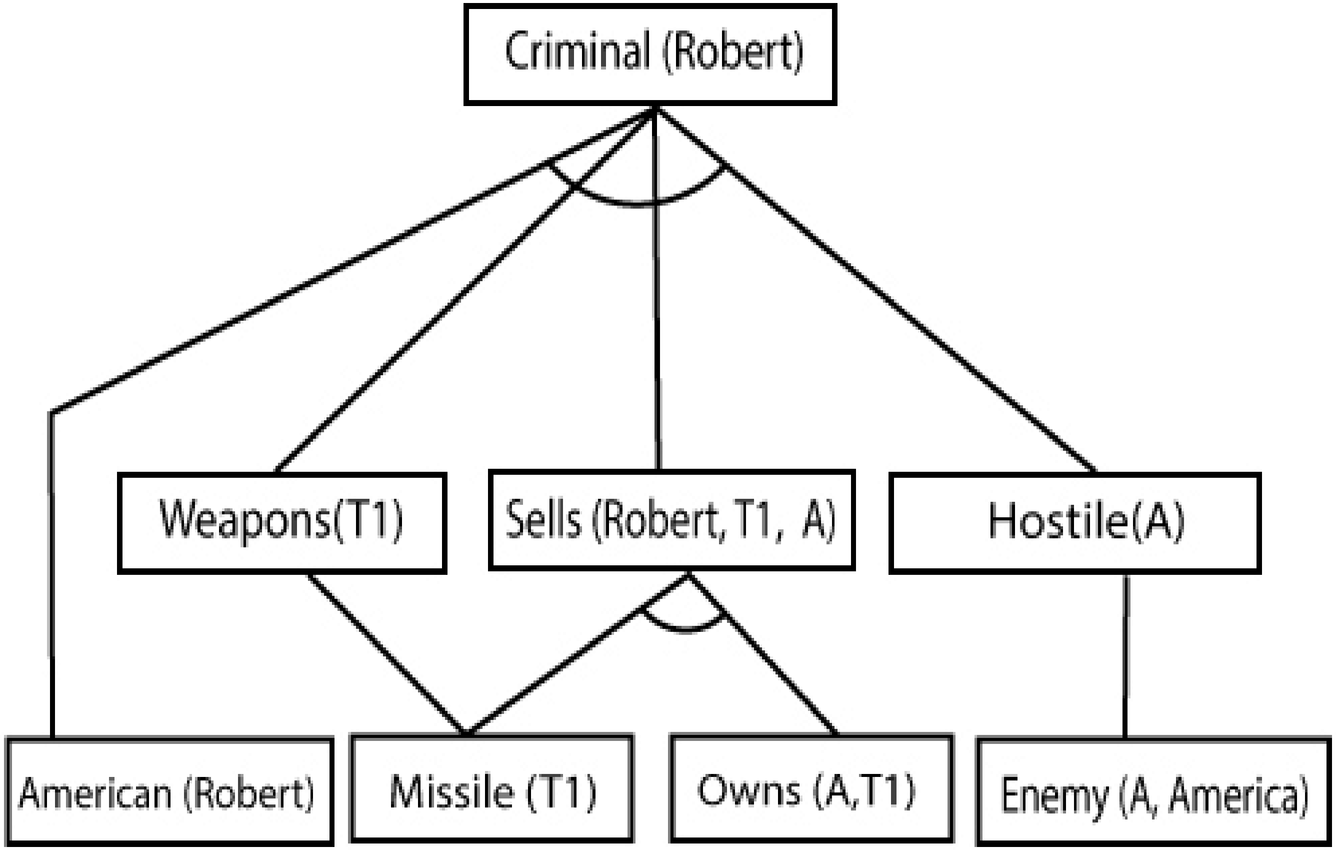
Missile (T1)

Owns (A,T1)

Enemy (A, America)

Forward chaining proof

- Step-3: Hence it is proved that Robert is Criminal



Backward Chaining

Also known as a backward deduction or backward reasoning method .

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.

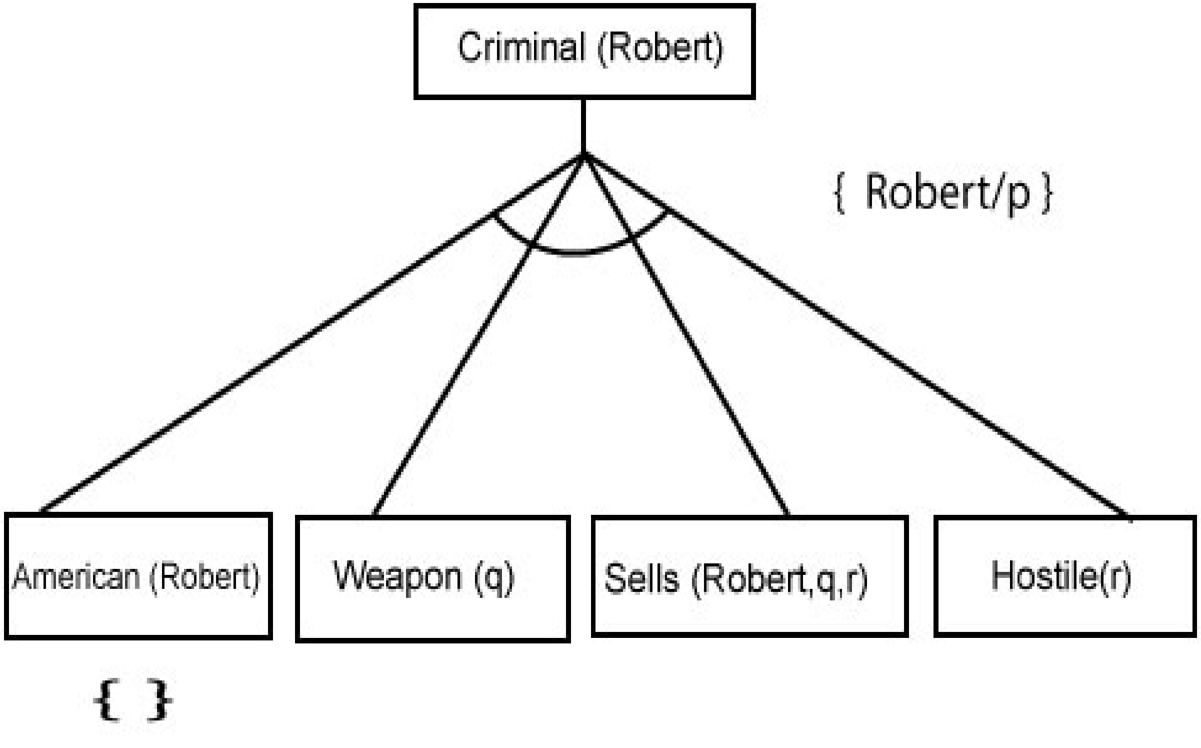
Properties of Backward Chaining

- 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.

Backward Chaining : Example

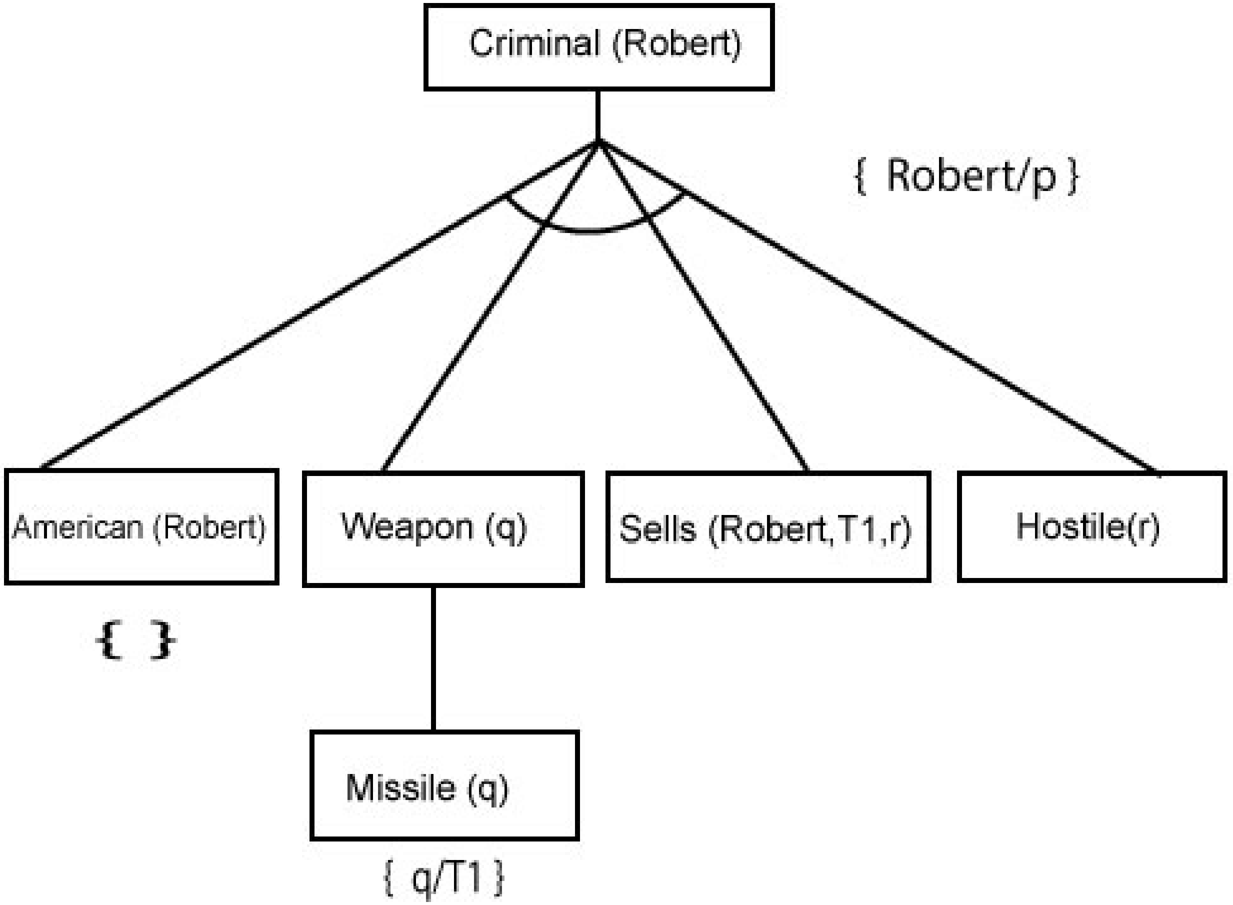
- **Step 1 :** Criminal (Robert)

- **Step 2 :**



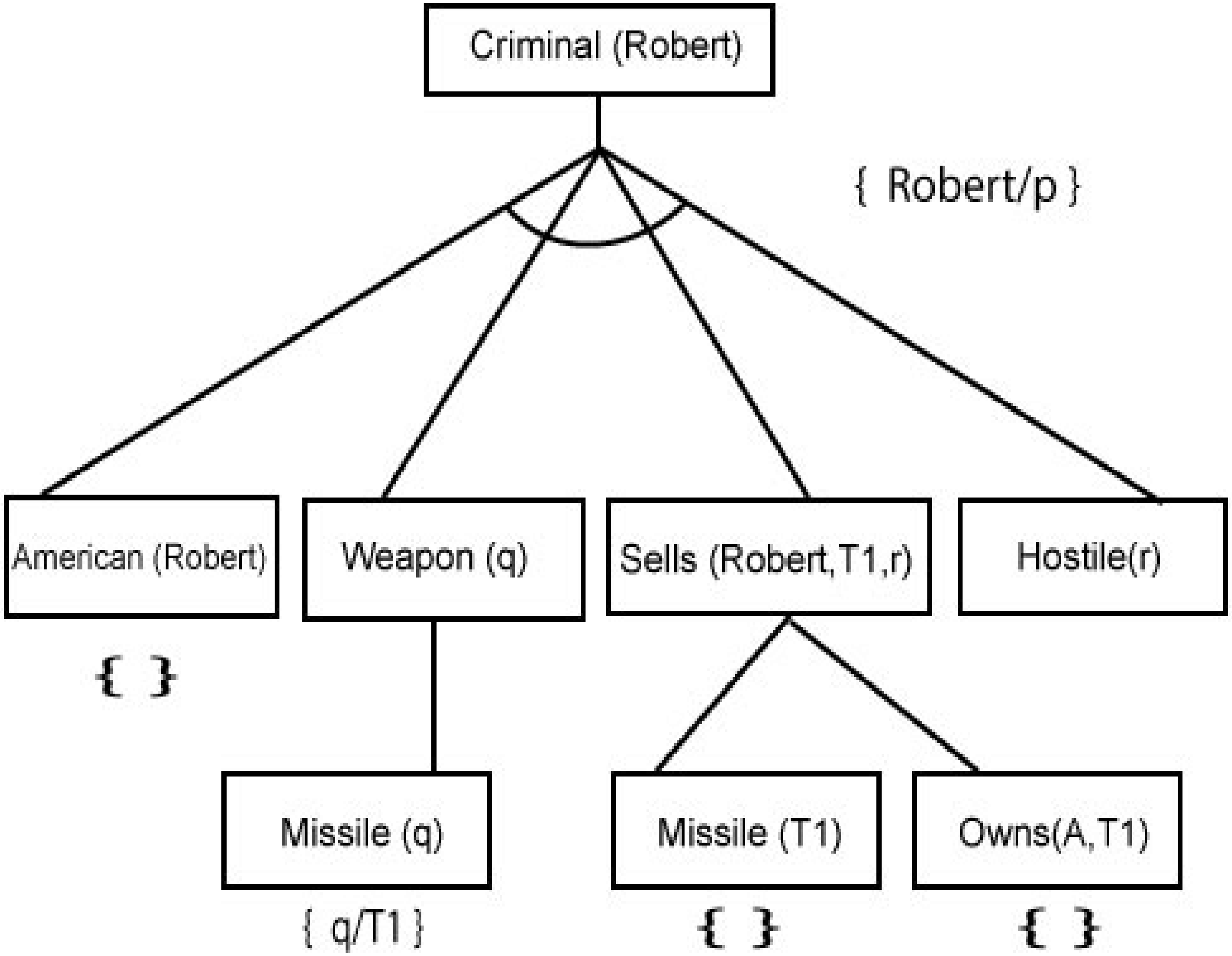
Backward Chaining : Example

- **Step 3 ·**



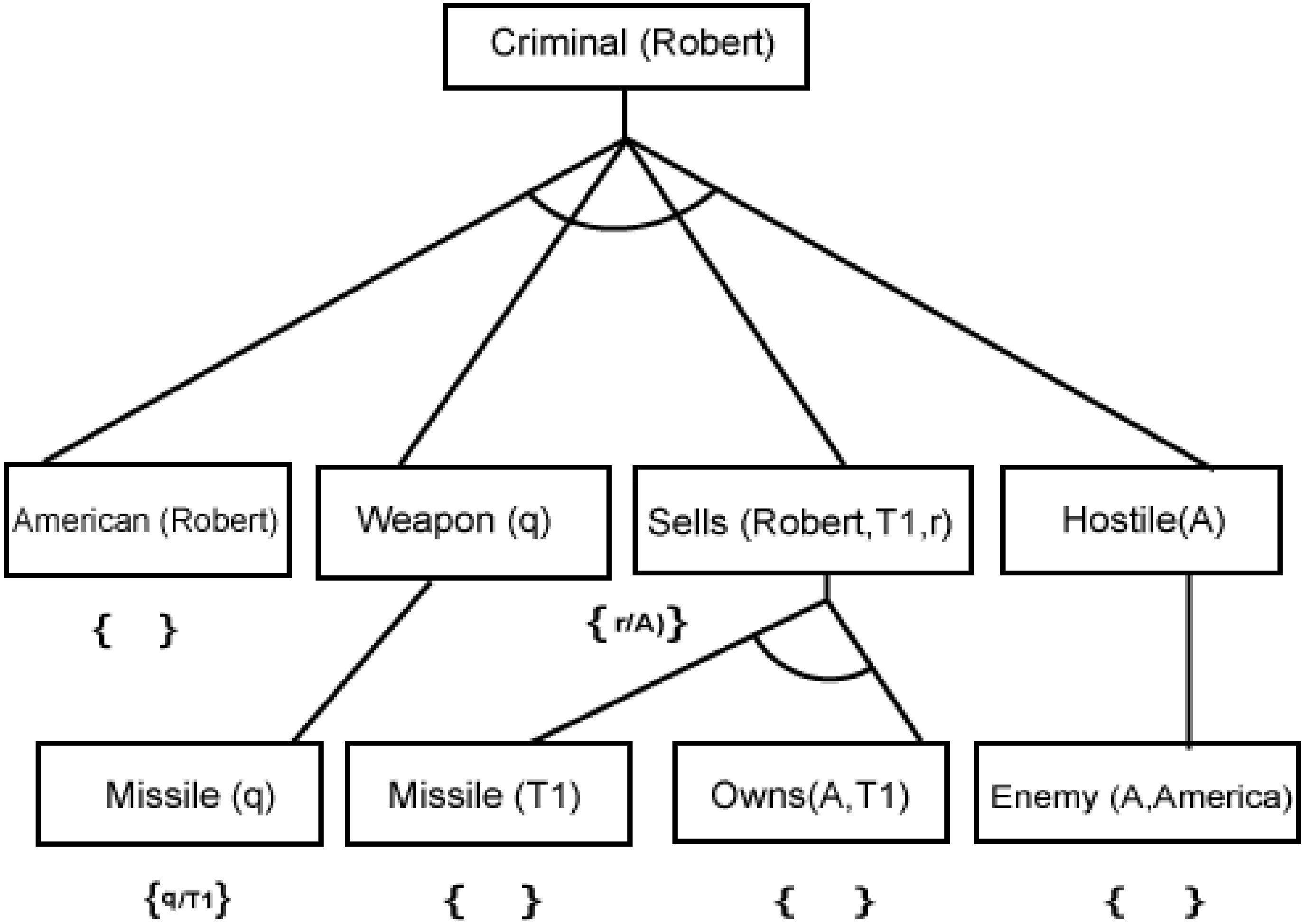
Backward Chaining : Example

- **Step 4 :**



Backward Chaining : Example

- **Step 5**



true

Difference between backward chaining and forward chaining

1.	Forward chaining starts from known facts and applies inference rule to extract more data unit it reaches to the goal.	Backward chaining starts from the goal and works backward through inference rules to find the required facts that support the goal.
2.	It is a bottom-up approach	It is a top-down approach
3.	Forward chaining is known as data-driven inference technique as we reach to the goal using the available data.	Backward chaining is known as goal-driven technique as we start from the goal and divide into sub-goal to extract the facts.
4.	Forward chaining reasoning applies a breadth-first search strategy.	Backward chaining reasoning applies a depth-first search strategy.

Difference between backward chaining and forward chaining

5.	Forward chaining tests for all the available rules	Backward chaining only tests for few required rules.
6.	Forward chaining is suitable for the planning, monitoring, control, and interpretation application.	Backward chaining is suitable for diagnostic, prescription, and debugging application.
7.	Forward chaining can generate an infinite number of possible conclusions.	Backward chaining generates a finite number of possible conclusions.
8.	It operates in the forward direction.	It operates in the backward direction.
9.	Forward chaining is aimed for any conclusion.	Backward chaining is only aimed for the required data.