Chapter 4

2020-09-09

Knowledge Representation, Inference & Reasoning

Er. Shiva K. Shrestha (HoD), Dept. of Computer Engineering (KhCE)

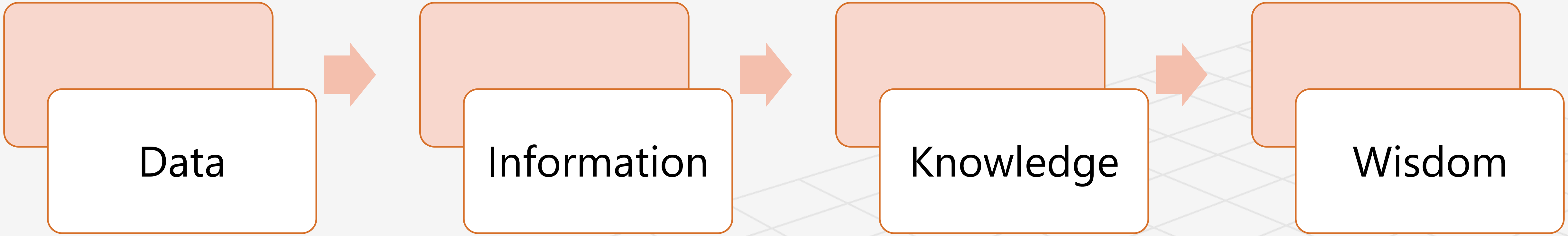
Chapter 4 – Knowledge Representation, Inference & Reasoning (8 Hrs.)

4.1. Formal Logic - Connectives, Truth Tables, Syntax, Semantics, Tautology, Validity, Well-formed-formula,

4.2. Propositional Logic, Predicate Logic, FOPL, Interpretation, Quantification, Horn-clauses,

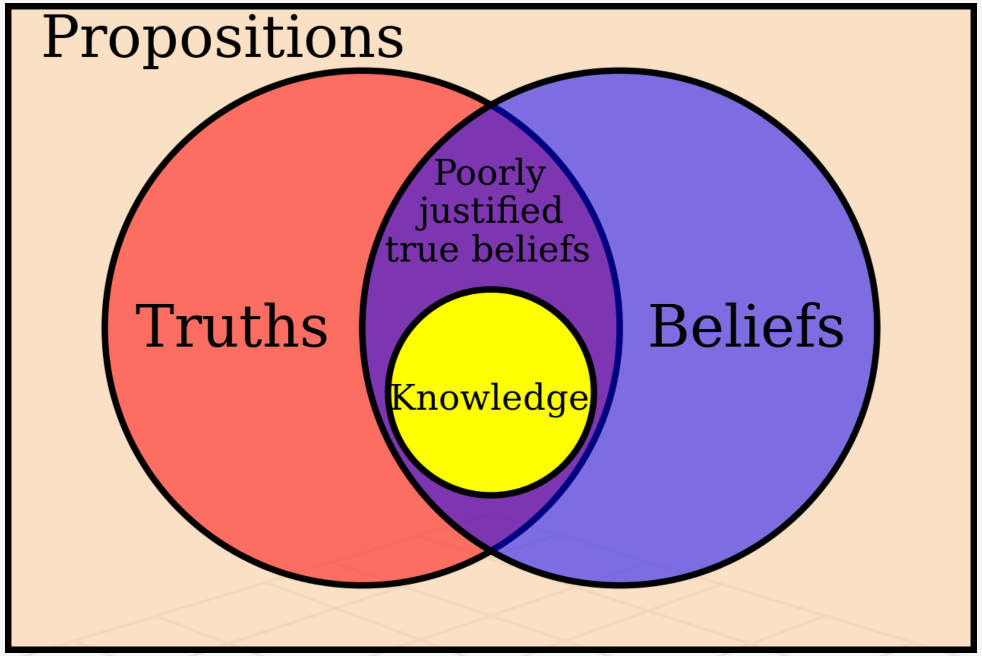
4.3. Rules of Inference, Unification, Resolution Refutation System (RRS), Answer Extraction from RRS, Rule Based Deduction System,

4.4. Statistical Reasoning - Probability and Bayes' Theorem and Causal Networks, Reasoning in Belief Network



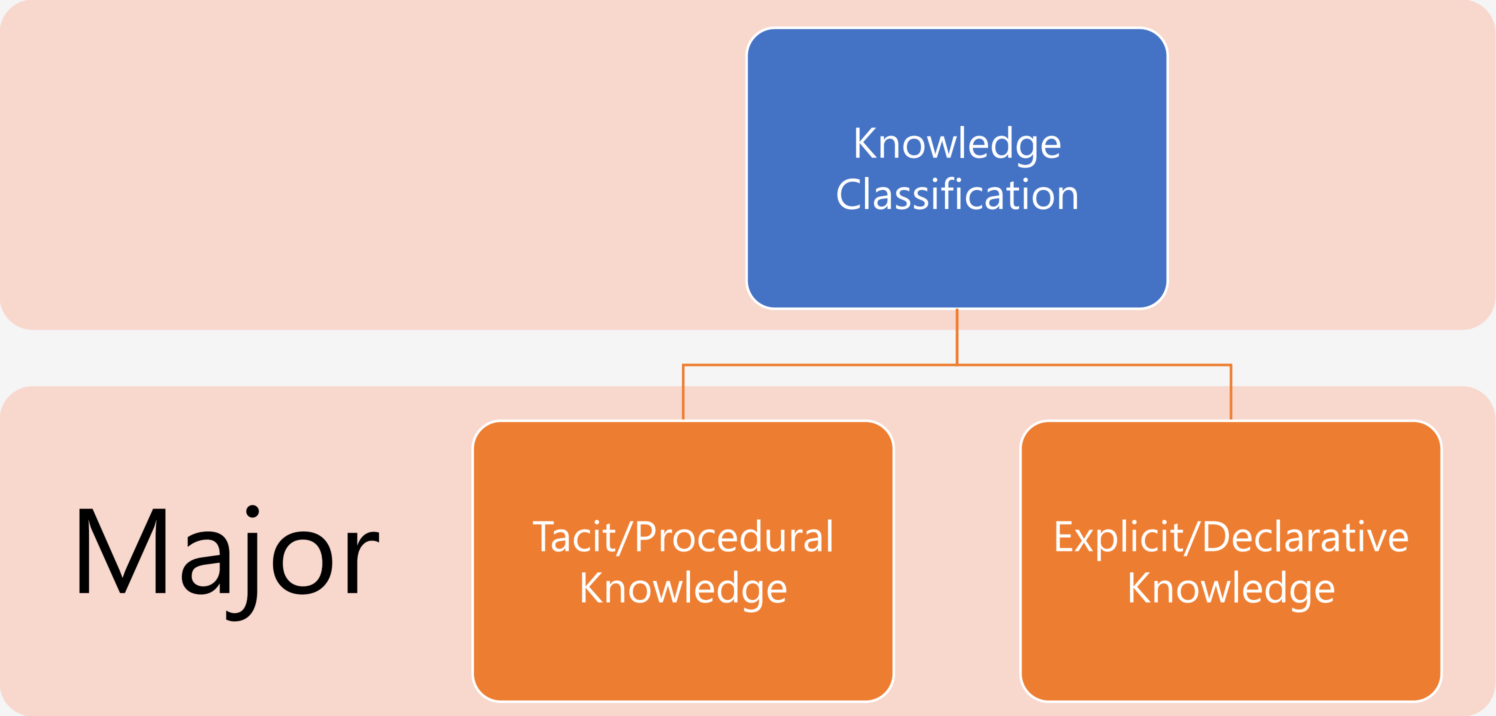
Knowledge

Knowledge - facts, information, and skills acquired through experience or education; the theoretical or practical understanding of a subject.



An Euler Diagram representing the traditional definition of knowledge as justified true belief.

Major Classification of Knowledge

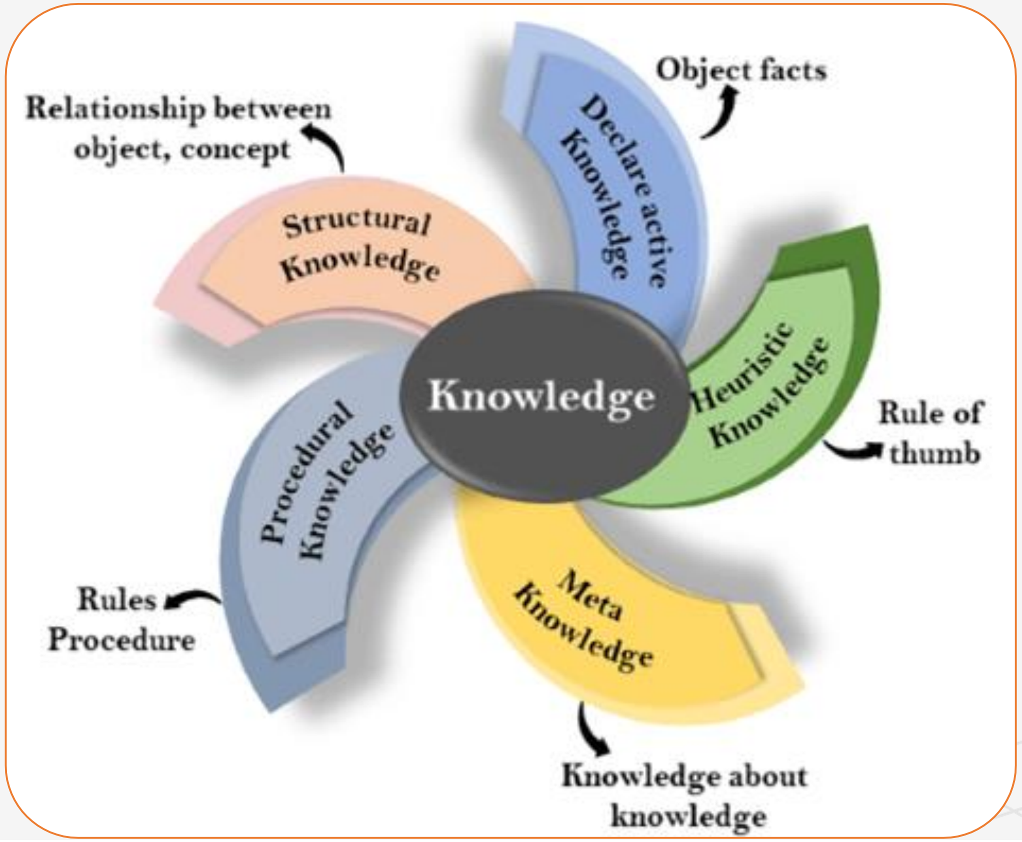


Other Sources of Knowledge

1. Inheritable Knowledge
2. Inferential Knowledge
3. Relations Knowledge
4. Heuristic Knowledge
5. Commonsense Knowledge
6. Explicit Knowledge
7. Uncertain Knowledge

Types of Knowledge

Knowledge is awareness or familiarity gained by experiences of facts, data, & situations



Type of Knowledge

Procedural/Operational Knowledge

- Includes rules, strategies, procedures, agendas, etc
- How to make coffee?

Declarative/Relational Knowledge

- Includes concepts, facts, & objects
- It is not hot today.

Structural Knowledge

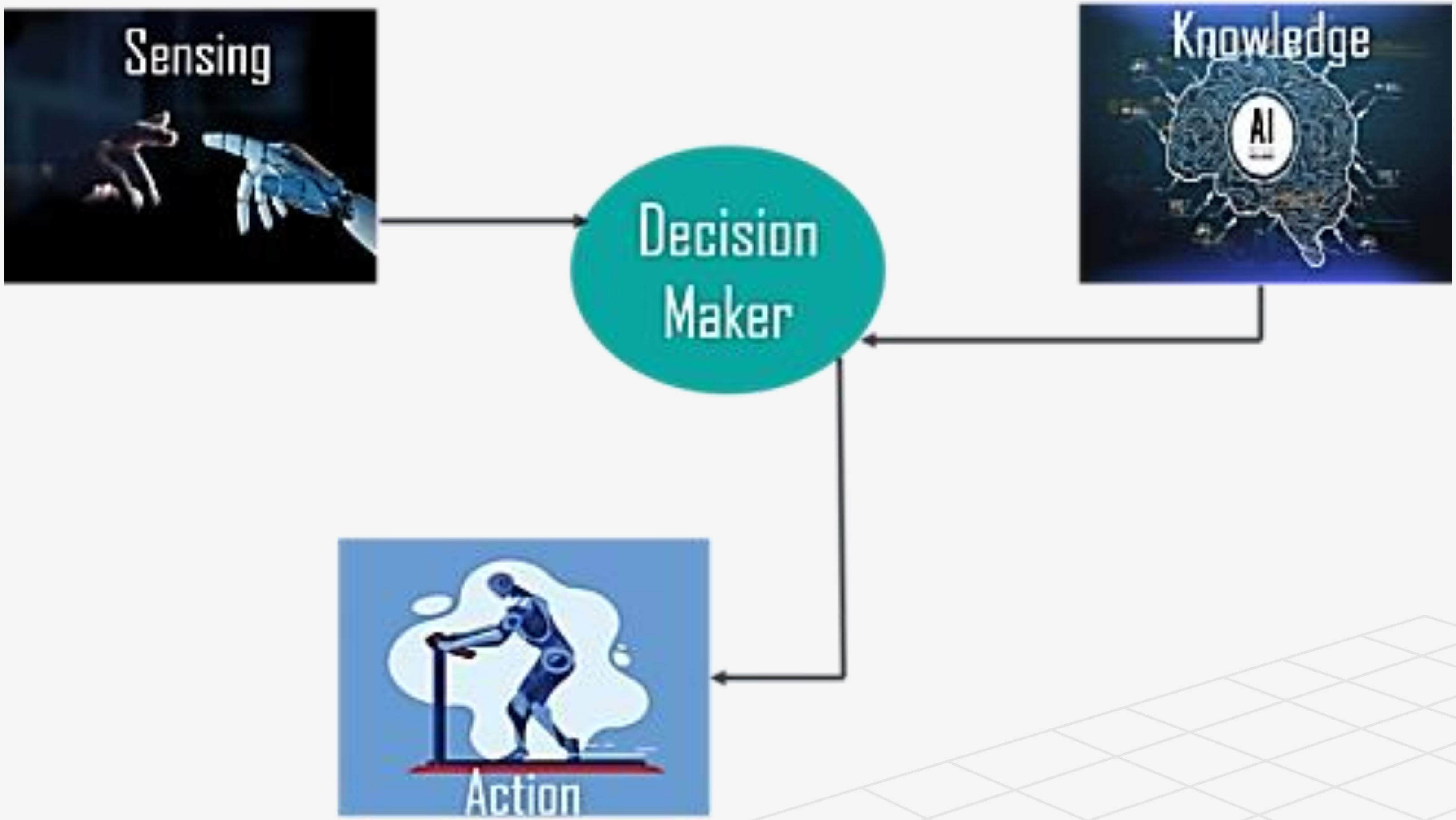
- Kind of, Part of, & Grouping of Sth
- How sys. components are joined together?

Heuristic Knowledge

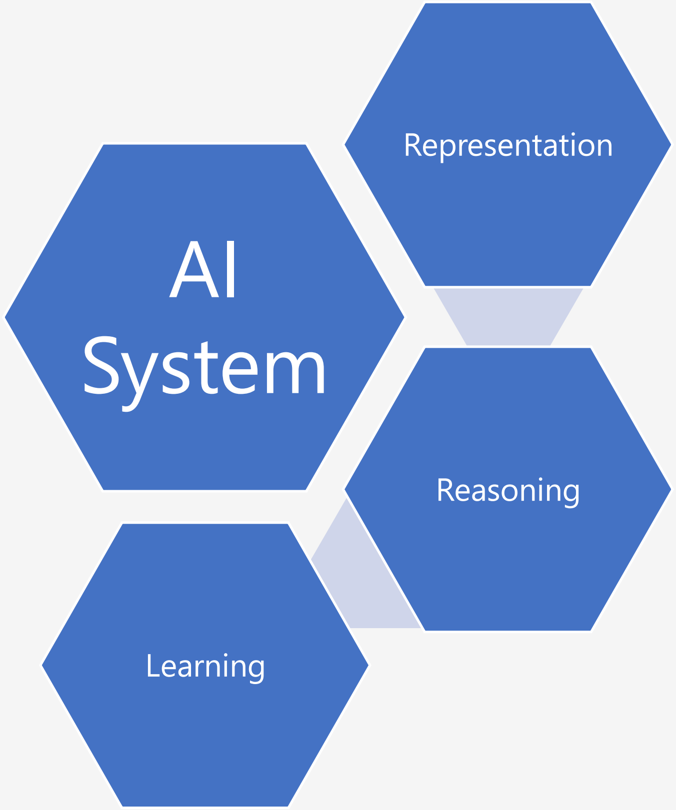
- Rules of thumb
- Drawing Conclusion

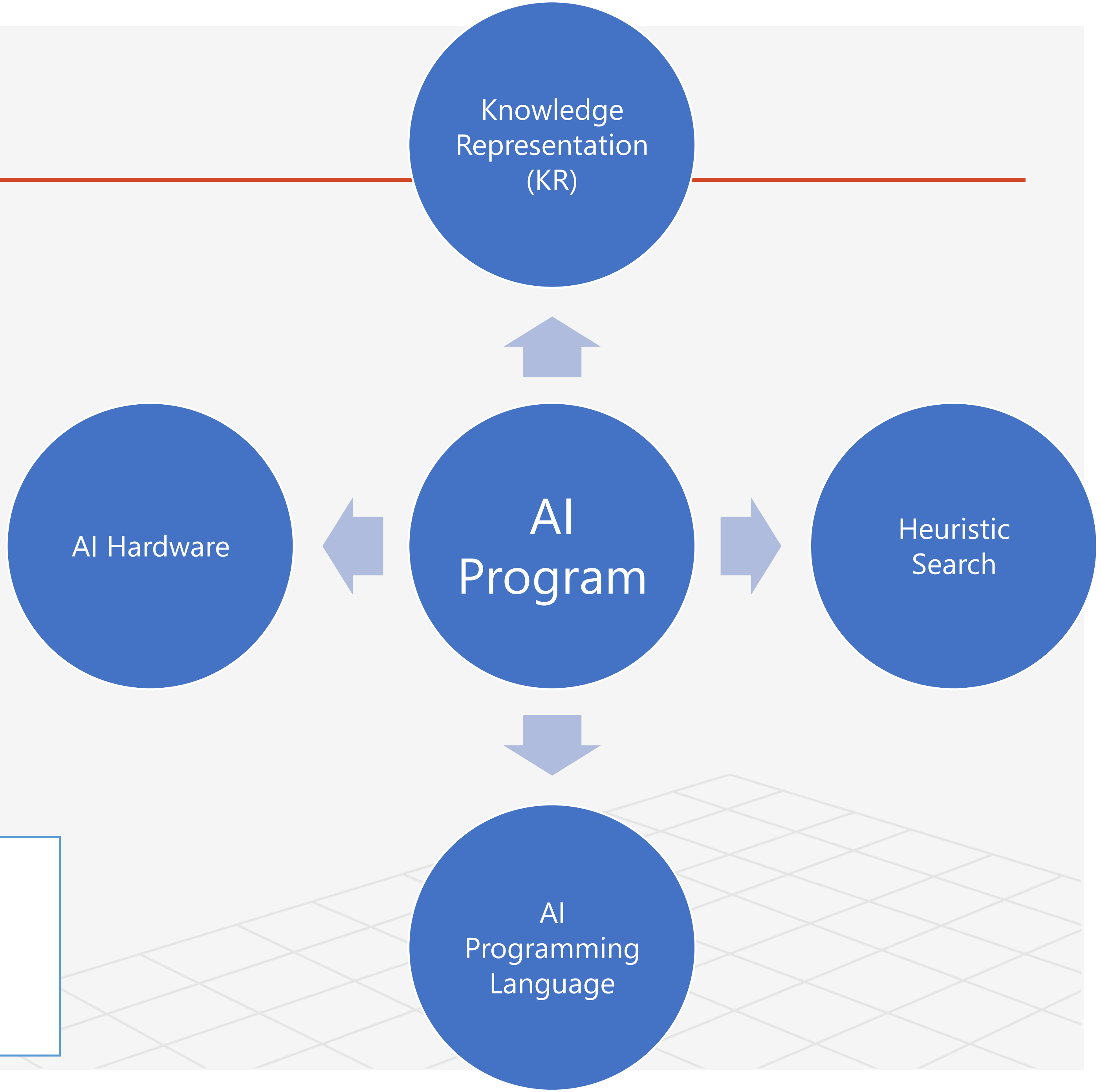
- Knowledge about Knowledge
- Blood Pressure Measure

Relationship Between Knowledge & Intelligence



AI System

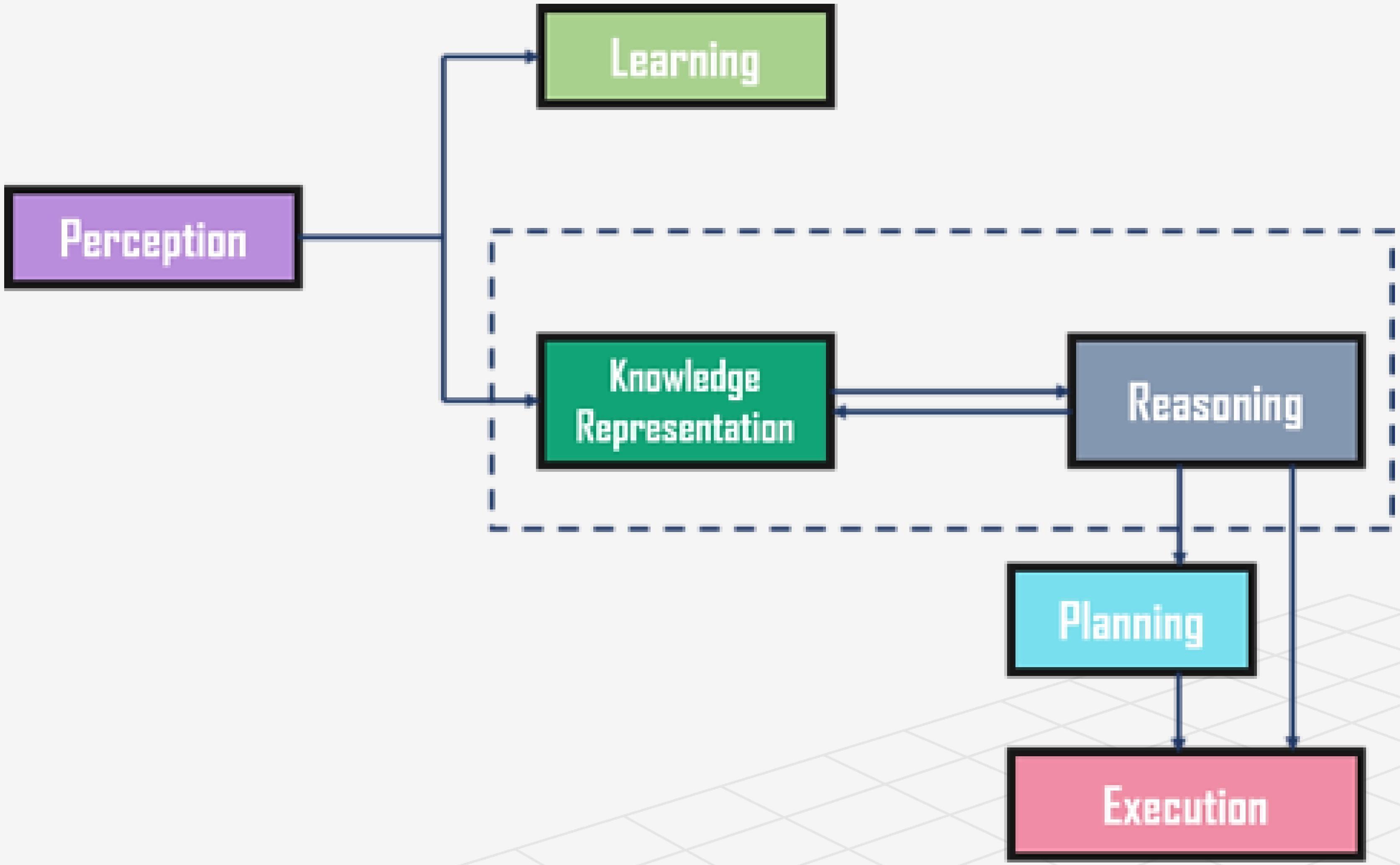




AI System must be capable of doing 3 things:

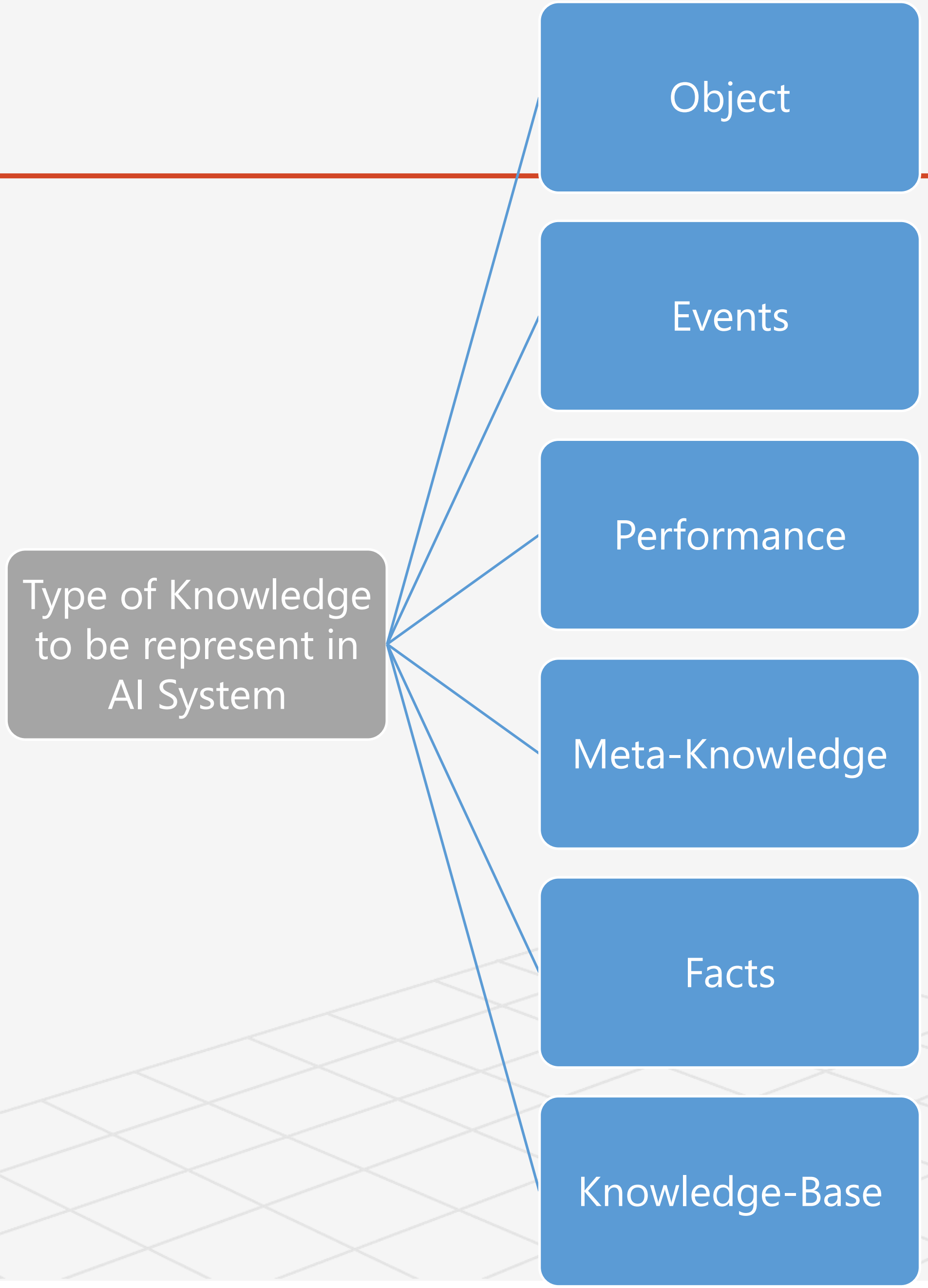
1. Store Knowledge
2. Apply knowledge stored to solve problems
3. Acquire new knowledge through experience

AI Knowledge Cycle



Knowledge Representation (KR)

- KR is not just storing data into some database, but it also *enables an intelligent machine to learn from that knowledge and experiences* so that *it can behave intelligently like a human*.
- **Knowledge Representation & Reasoning (KRR)** is the *part of AI* which concerned with AI agents thinking & how thinking contributes to intelligent behavior of agents.



Facts & Rules

Facts can be classified as

- Single Valued or Multivalued Facts
- Uncertain Facts
- Fuzzy Facts

Rules can be classified as

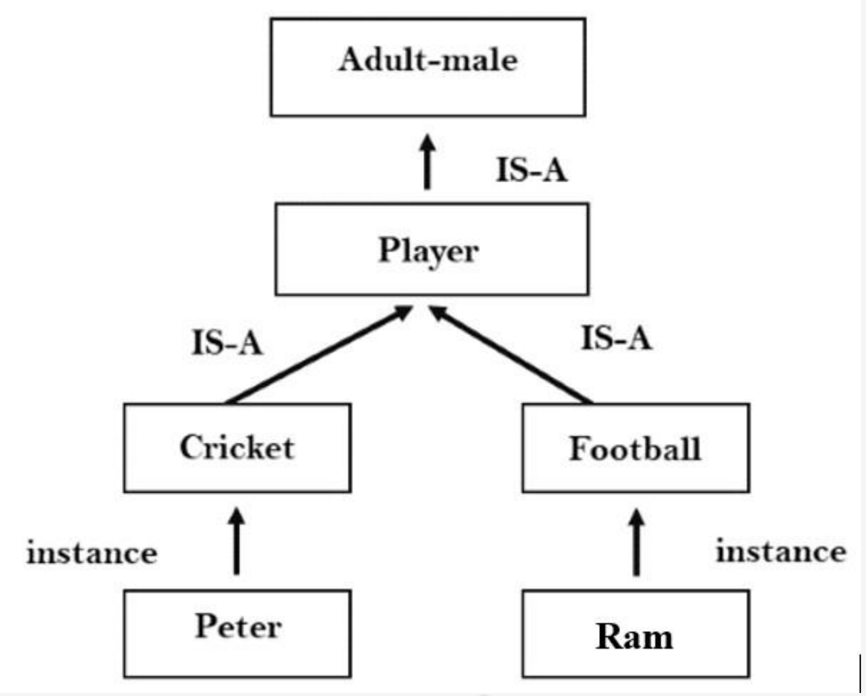
- Relationships
- Recommendations
- Directive
- Variable
- Uncertain Rules
- Meta Rules

Approaches to KR

1. Simple Relational Knowledge
2. Inheritable Knowledge
3. Inferential Knowledge
4. Procedural Knowledge

If-Then rule

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



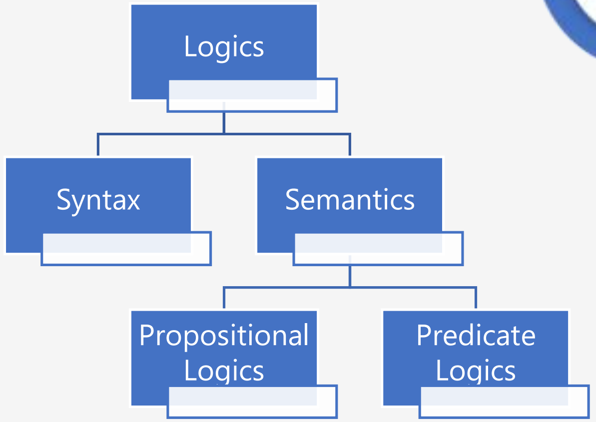
- Marcus is a man
- All men are mortal

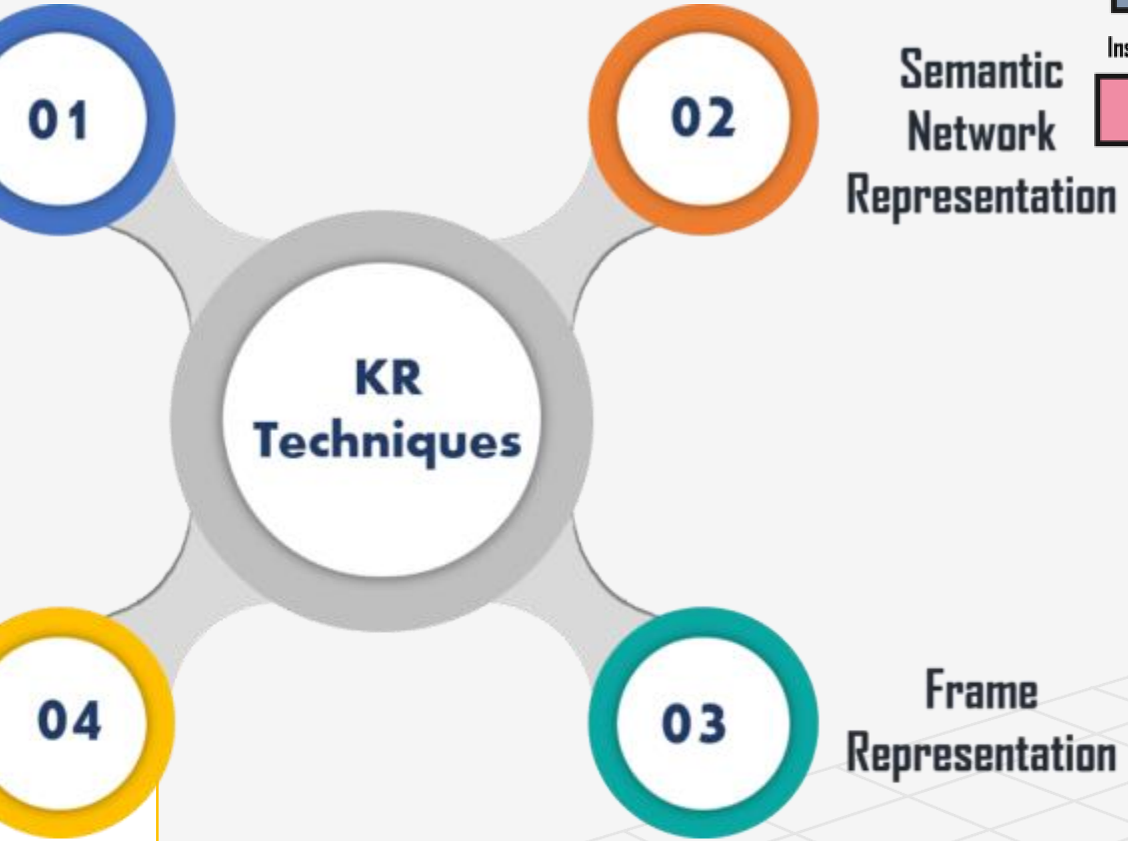
$$man(Marcus)$$

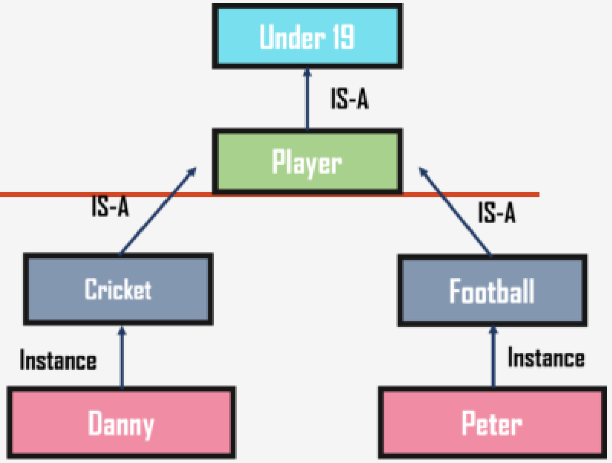
$$\forall_x : man(x) \rightarrow mortal(x)$$

KR Techniques

Name	Age	Emp ID
Shakti	25	100071
Abhaya	23	100056
Shyam	27	100042



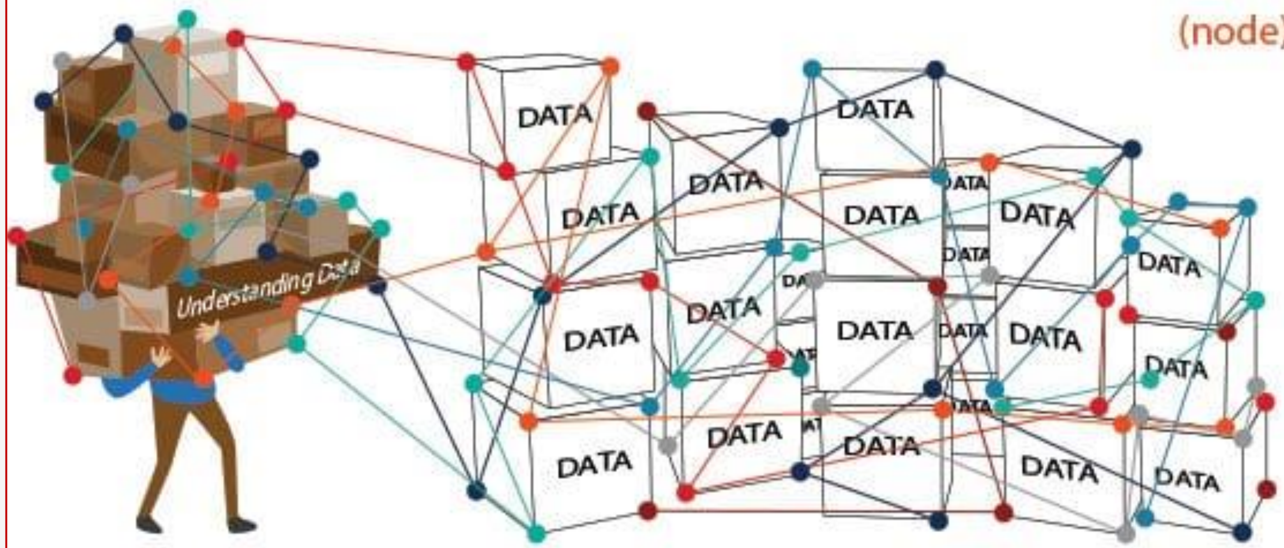


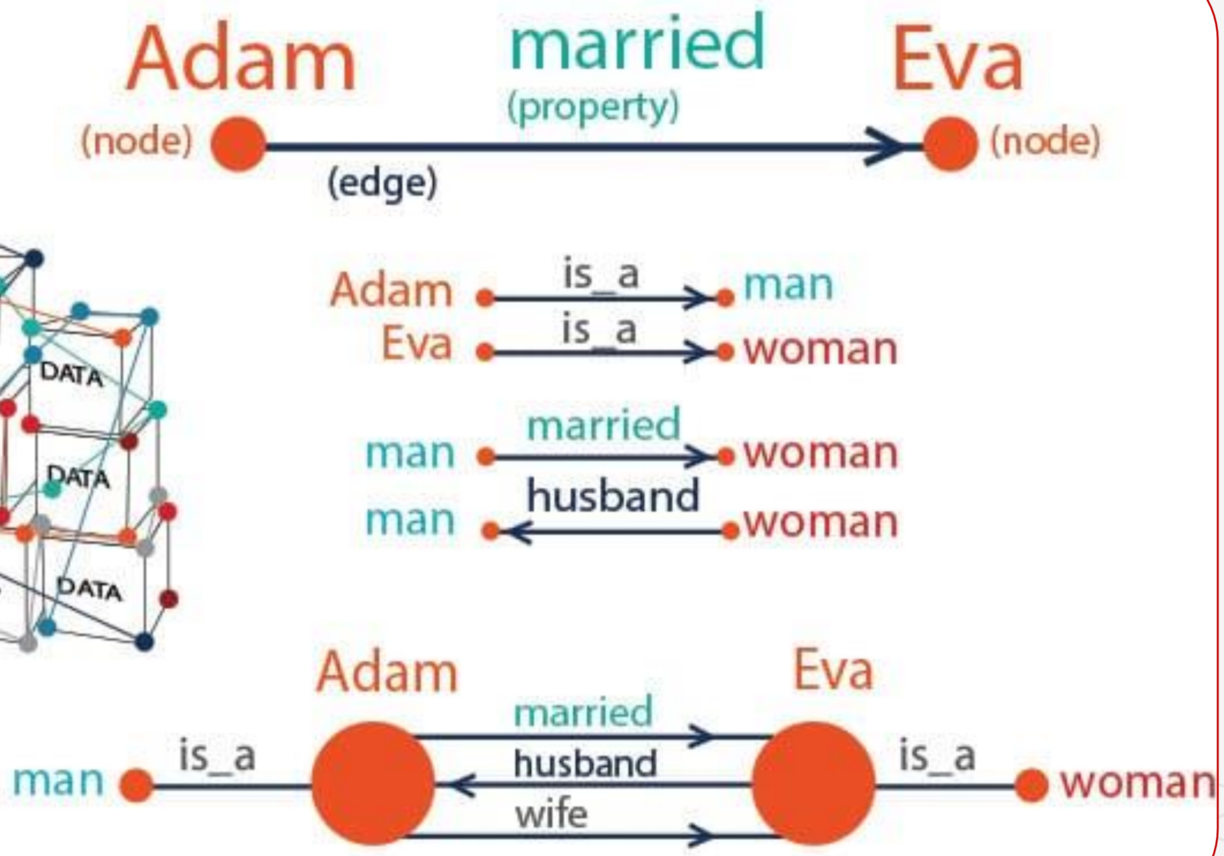


Main Three Parts:
1. The set of production rules
2. Working Memory
3. The recognize-act-cycle

Slots	Filters
Title	Artificial Intelligence
Genre	Computer Science
Author	Peter Norvig
Edition	Third Edition
Year	1996
Page	1152

Knowledge Base





Knowledge Base System (KBS)

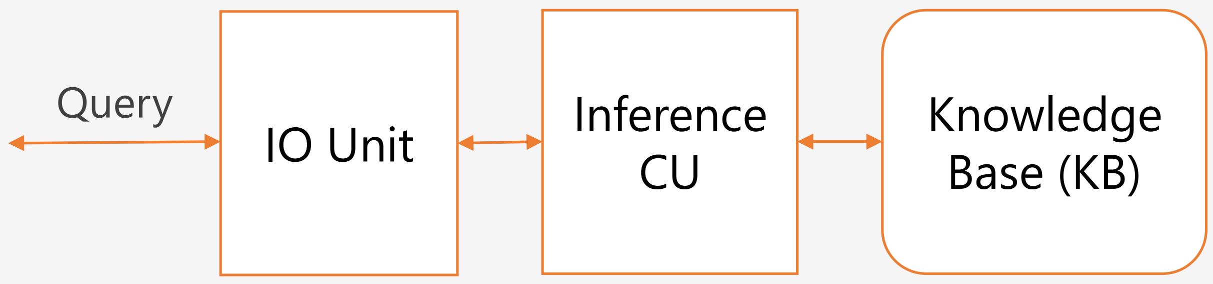


Fig.: Components of KBS

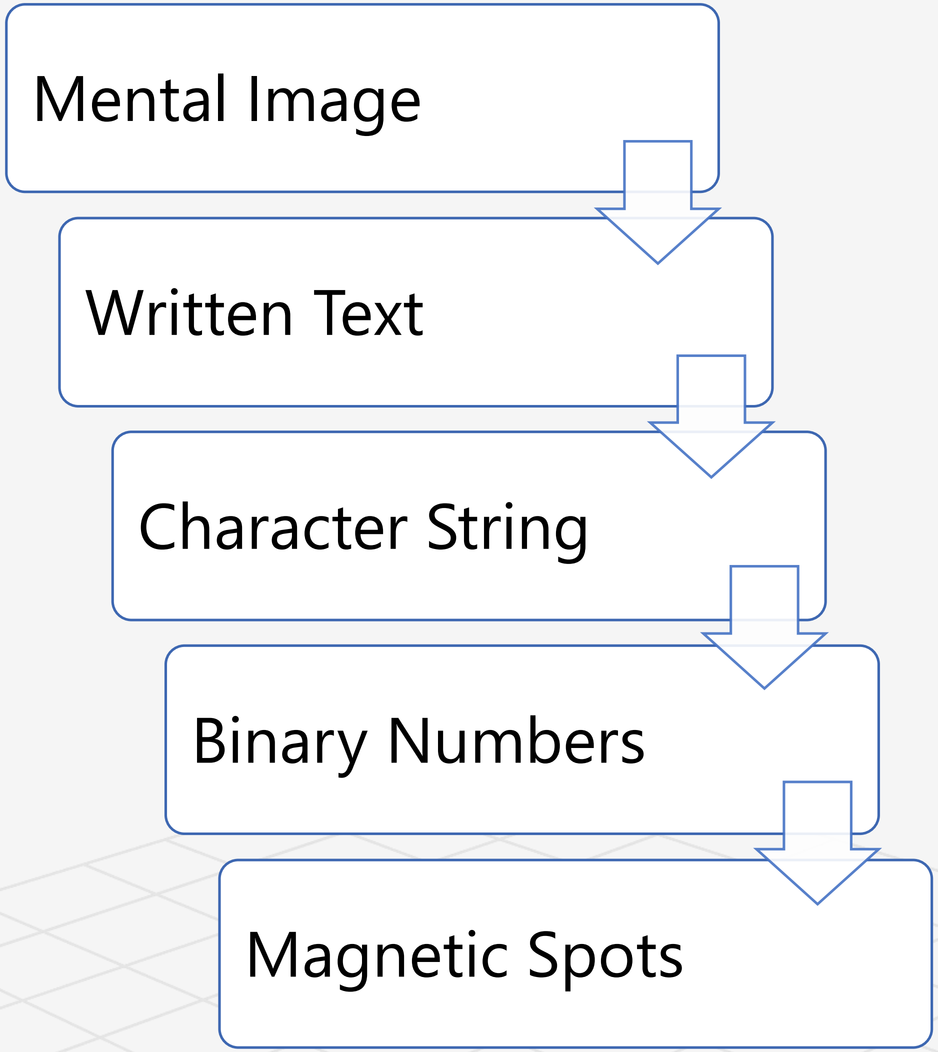
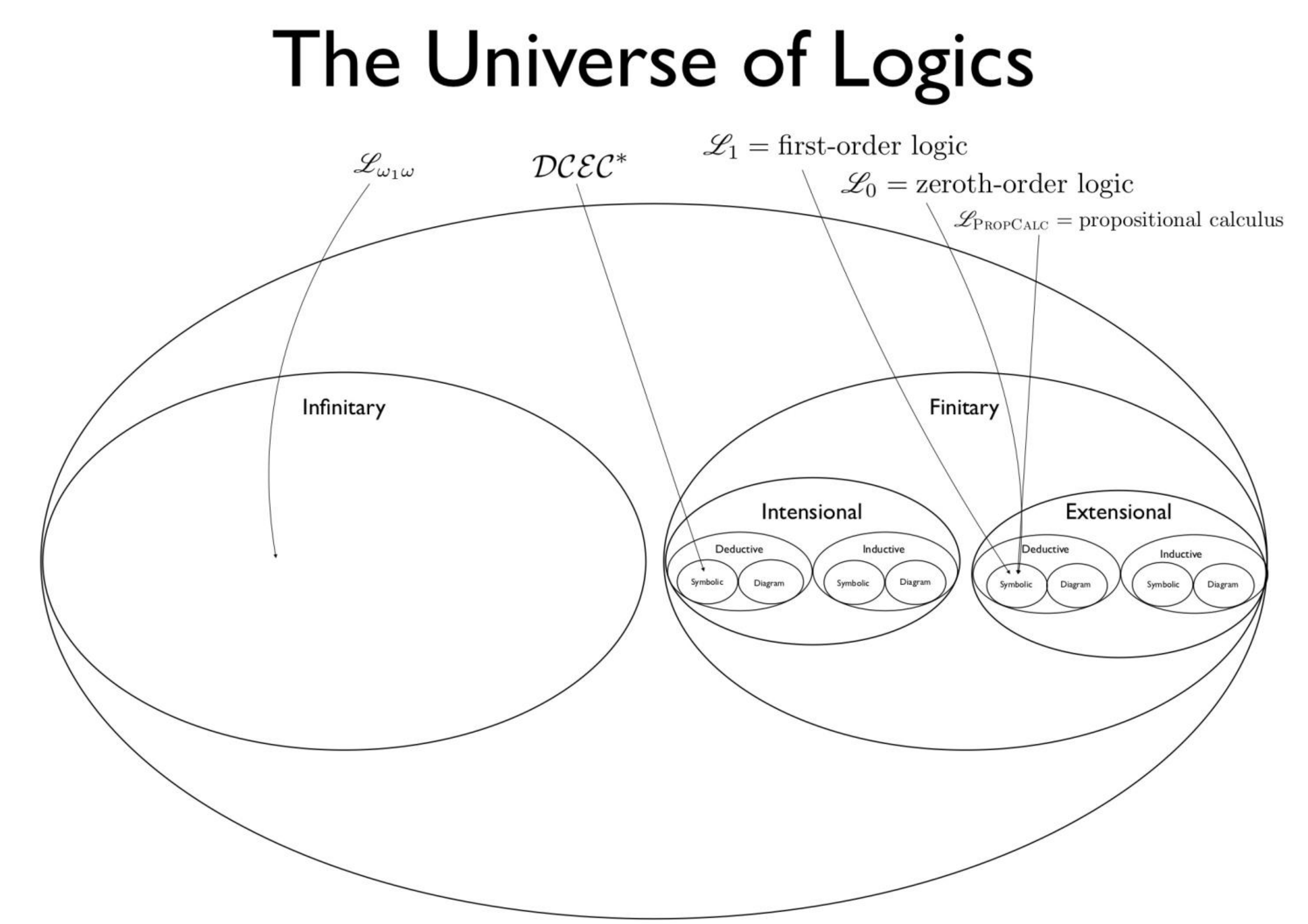


Fig.: Levels of KR

4.1. Formal Logic - Connectives, Truth Tables, Syntax, Semantics, Tautology, Validity, Well-formed-formula

Formal Logic - Connectives, Truth Tables, Syntax, Semantics, Tautology, Validity, Well-formed-formula



Formal Logic Terms

Logic – Formal language for representing knowledge such that conclusions can be draws.

Proposition – Collection of declarative statements that has either a truth or false value.

Tautology – It is a formula or assertion that is *true in every possible interpretation*.

Contradiction – The propositions that are *always false* is called contradiction.

Contingency – A proposition that is *nor a contradiction* is called a contingency.

Predicate Calculus – Deals with predicates, which are propositions containing variables.

Quantifier – It is a language element that helps in generation of a quantification.

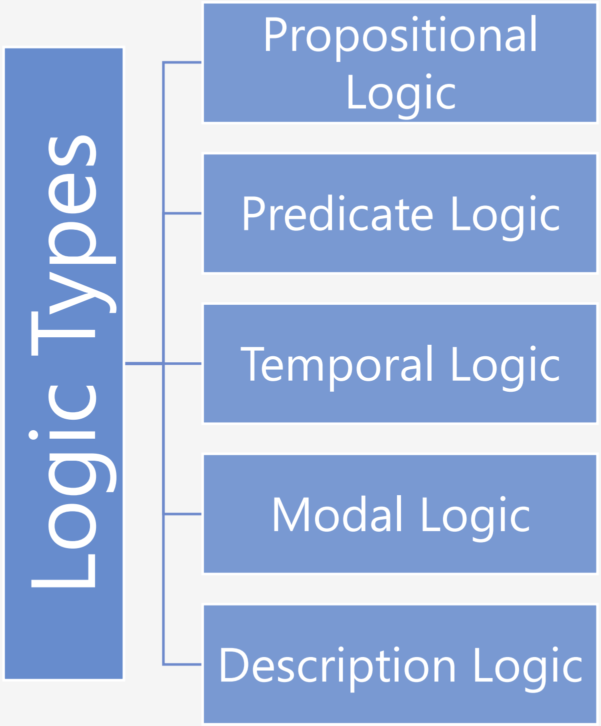
Universal Quantifier – Logical constant which is interpreted as "given any" or "for all"

Existential Quantifier – Logical constant which is interpreted as "there exists", "there is at least one", or "for some".

First Order Predicate Logic (FOPL) – Predicate Logic, Quantificational Logic, is a collection of formal systems used in mathematics, philosophy, linguistics, and computer science.

4.2. Propositional Logic, Predicate Logic, FOPL, Interpretation, Quantification, Horn-clauses

Propositional Logic, Predicate Logic, FOPL, Interpretation, Quantification, Horn-clauses



Types of formal mathematical logic

Propositional logic

- Propositions are interpreted as true or false
- Infer truth of new propositions

First order logic

- Contains predicates, quantifiers and variables
- E.g. $philosopher(x) \rightarrow scholar(a)$
- $\forall x: king(x) \wedge greedy(x) \rightarrow evil(x)$
- Variables range over individuals (domain of discourse)

Second order logic

- Quantify over predicates and over sets of variables

Other Logics

Temporal logic

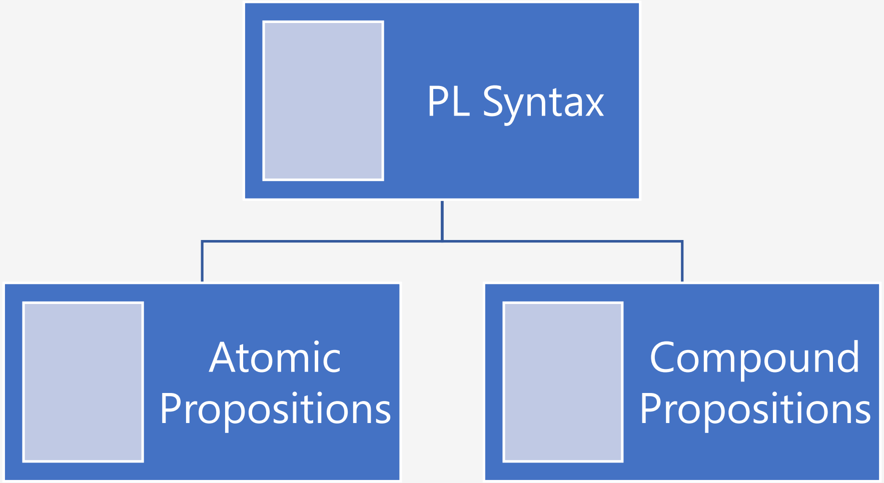
- Truths & relationships change & depend on time

Fuzzy logic

- Uncertainty, contradictions

Propositional Logic (PL)

- It is also called Boolean Logic as it works on 0 and 1.
- PL is the simplest form of logic where all the statements are made by propositions.
- A proposition is a declarative st. which is either true or false.
- It is a technique of KR in logical & mathematical form.



Atomic Propositions Example:
a) It is Sunday.
b) The Sun rises from West (False proposition)
c) 3+3= 7(False proposition)
d) 5 is a prime number.

Compound Propositions Example:
a) “It is raining today, and street is wet.”
b) “Govinda is a doctor, and his clinic is in Kathmandu.”

PL Terminology

1. Statement
   a. Satisfiability
   b. Contradiction or Unsatisfiable
   c. Validity $P \vee \sim P = T$
   d. Logical Equivalence
2. Conjunctive or Operator
3. Truth Value
4. Tautologies $P \vee \sim P = T$
5. Contradictions $P \wedge \sim P = T$
6. Contengencies $P \vee Q = T$
7. Antecedent & Consequent $P \rightarrow Q$
   - P is 'if-clause', antecedent
   - Q is then clause, Consequent
8. Argument
   - Compound Statement

Logical Connectives

Connective Symbols	Word	Technical Term	Example
$\neg$	Not	Negation	$\neg A$ *or* $\neg B$
$\wedge$	AND	Conjunction	$A \wedge B$
$\vee$	OR	Disjunction	$A \vee B$
$\rightarrow$	Implies	Implication	$A \rightarrow B$
$\leftrightarrow$	If & Only if	Bi-conditional	$A \leftrightarrow B$

Truth Table

Logical Equivalence

A	B	¬A	A ∧ B	¬A ∨ B	A → B
T	T	F	T	T	T
T	F	F	F	F	F
F	T	T	F	T	T
F	F	T	F	T	T

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

Figure 7.8 Truth tables for the five logical connectives. To use the table to compute, for example, the value of $P \vee Q$ when P is true and Q is false, first look on the left for the row where P is *true* and Q is *false* (the third row). Then look in that row under the $P \vee Q$ column to see the result: *true*.

Precedence	Operators
1st Precedence	Parenthesis
2nd Precedence	Negation
3rd Precedence	Conjunction
4th Precedence	Disjunction
5th Precedence	Implication
6th Precedence	Bi-conditional

Task

1. Prove that $-\big((P \wedge Q) \rightarrow R\big) \vee (\sim Q \rightarrow \sim R)$ is valid.

2. Show that the set of statements "I will be wet if it rains and I go out of the house. It is raining now. I go out of the house. I will not be wet" are inconsistent.

Well Formed Formula (WFF)

- An expression is referred to as a WFF if it is constructed correctly according to the rules of syntax of propositional calculus.
- WFF consists of atomic symbols joined with connectives.

  $\therefore P, P \wedge \sim P, P \wedge Q, P \vee Q, P \rightarrow Q, P \leftrightarrow Q$
- A WFF is defined recursively as
  1. An atom (say P) is a WFF.
  2. If P is a WFF then $\sim P$ is a WFF.
  3. If P & Q are WFF, then $P \wedge Q, P \vee Q, P \rightarrow Q, P \leftrightarrow Q$ are WFF.
  4. A string of symbols is a WFF if and only if it is obtained by using above rule 1 to rule 3.

Set of Inference Rules & Laws

Rule 1: Idempotency Rule

Rule 2: Commutative Law

Rule 3: Associative Law

Rule 4: Distributive Law

Rule 5: De Morgan's Law

Rule 6: Implication Removal

Rule 7: Bi-conditional Elimination

Rule 8: Absorption Rule

Rule 9: Contrapositive

Rule 10: Double Negation

Rule 11: Fundamental Identities

Rule 12: AND Rule (Introduction)

Rule 13: AND Rule (Elimination)

Rule 14: OR Rule (Introduction)

Rule 15: Modus Ponens

Rule 16: Modus Tollens

Rule 17: Hypothetical Syllogism/Chain Rule

Rule 18: Reductio Ad Absurdum

Rule 19: Disjunctive Syllogism

Rule 20: Constructive & Destructive Dilemma

Standard Logical Equivalences

$(\alpha \wedge \beta)$	$\equiv$	$(\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv$	$(\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv$	$(\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv$	$(\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv$	$(\neg\alpha \vee \neg\beta)$	De Morgan
$\neg(\alpha \vee \beta)$	$\equiv$	$(\neg\alpha \wedge \neg\beta)$	De Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

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

$p \rightarrow q \equiv \neg p \vee q$	Relation by Implication (RBI)
$p \rightarrow q \equiv \neg q \rightarrow \neg p$	Contraposition
$p \Leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$	Definition of Biconditional
$p \oplus q \equiv (p \vee q) \wedge \neg(p \wedge q)$	Alternate Definition of xor

TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

Set of Inference Rules & Laws

20 Rules? – Let’s see Lecturer Note

Type of Inference Rules

- Modus Ponens: $\dfrac{P \rightarrow Q,\quad P}{\therefore Q}$
- Modus Tollens: $\dfrac{P \rightarrow Q,\quad \sim Q}{\sim P}$
- Hypothetical Syllogism: $\dfrac{P \rightarrow Q \qquad Q \rightarrow R}{P \rightarrow R}$
- Disjunctive Syllogism: $\dfrac{P \vee Q,\quad \neg P}{Q}$
- Addition: $\dfrac{P}{P \vee Q}$
- Simplification: $\dfrac{P \wedge Q}{Q}$ Or $\dfrac{P \wedge Q}{P}$
- Resolution: $\dfrac{P \vee Q,\quad \neg P \wedge R}{Q \vee R}$

Inference Rules Terms

Some terminologies related to inference rules:

- **Implication:** It is one of the logical connectives which can be represented as $P \rightarrow Q$. It is a Boolean Expression.
- **Converse:** The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as $Q \rightarrow P$.
- **Contrapositive:** Negation of converse is termed as contrapositive, & it can be represented as $\neg Q \rightarrow \neg P$.
- **Inverse:** The negation of implication is called inverse. It can be represented as $\neg P \rightarrow \neg Q$.

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	$\neg Q \rightarrow \neg P$	$\neg P \rightarrow \neg Q$.
T	T	T	T	T	T
T	F	F	T	F	T
F	T	T	F	T	F
F	F	T	T	T	T

Inference Rules

Inference Rules

1. Modus Ponens
2. Modus Tollens
3. Hypothetical Syllogism
4. Disjunctive Syllogism
5. Addition
6. Simplification
7. Conjuction
8. Resolution

TABLE 1 **Rules of Inference.**

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore\ q$	$(p \land (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore\ \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore\ p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore\ q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p $\therefore\ p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ $\therefore\ p$	$(p \land q) \to p$	Simplification
p q $\therefore\ p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ $\therefore\ q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

Modus Ponens (MP) & Modus Tollens (MT)

Rule 1: Modus Ponens

" If Tweety is a bird then Tweety flies. "
" Tweety is a bird. "

Therefore, Tweety flies .

Notations for M.P.

$$\frac{P \rightarrow Q,\ P}{Q}$$

Rule 2: Module Tollens

" If Tweety is a bird then Tweety flies. "
" Tweety doesn't fly. "

Therefore, Tweety is not a bird .

Notations for M.T.

$$\frac{P \rightarrow Q,\ {\sim}Q}{{\sim}P}$$

Hypothetical Syllogism (HS) & Disjunctive Syllogism (DS)

Rule 3: Hypothetical Syllogism

"If you have my password, then you can log on to my facebook. "

" If you can log on my to facebook then you can delete my facebook account. "

Therefore, If you have my password then you can delete my facebook account.

Notations for H.S.

$$\frac{P \to Q,\ Q \to R}{P \to R}$$

Rule 3: Disjunctive Syllogism

"The ice cream is either vanilla flavored or chocolate flavored."

" The ice cream is not vanilla flavored. "

Therefore, The ice cream is chocolate flavored.

Notations for D.S.

$$\frac{P \lor Q,\ {\sim}P}{Q}$$

Task

1. Ram and Shyam is a student.
2. Today is Sunday or Monday.
3. It is not Sunday.
4. Ram teaches Shyam or Gita but not at a same time.
5. If it is hot then it shall rain.
6. If the humidity is high it will rain either today or tomorrow.
7. If the wind speed is high and the temperature is low, then one does not feel comfortable.
8. It requires courage and skills to climb a mountain.
9. Cancer will not be cured unless its causes is determined and a new drug for cancer is found.
10. He requires a doctor and also a lawyer if and only if he is sick also injured.

Predicate Logic or FOL or FOPL

Predicates allow us to talk about objects

- **Properties:** is_wet(today)
- **Relations:** likes(john, apples)
- True or false

In predicate logic each atom is a predicate

- e.g. first order logic, higher-order logic

Predicate Logic Expressions

- Logical Operators - &&, ||
- Quantifiers - < >
- Universal Quantifiers - ∀
- Existential Quantifiers - ∃

- PL assumes the world contains facts that are true or false.
- FOL assumes the world contains
  - **Objects:** people, houses, numbers, colors, baseball games, wars, …
  - **Relations between objects:** red, round, prime, brother of, bigger than, part of, comes between, …

FOL Examples

- $male(Ram)$
- $father(Dasarath, Ram)$
- $likes(Ram, Sita)$

FOL (2)

Relations

- Some relations are properties: they state some fact about a single object: Round(ball), Prime(7).
- n-ary relations state facts about two or more objects:

  Married(John,Mary),

  Largerthan(3,2).
- Some relations are functions: their value is another object: Plus(2,3), Father(Dan).

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

Quantification

- Round(ball) is true or false because we give it a single argument (ball).
- We can be much more flexible if we allow *variables* which can take on values in a domain. e.g. reals x, all persons P, etc.
- To construct logical sentences we need a quantifier to make it true or false.

Using FOL

- We want to TELL things to the KB, e.g. $TELL(KB, \forall x: king(x) \rightarrow person(x))$
- We also want to ASK things to the KB, $ASK(KB, \exists x: person(x))$
- The KB should return the list of $x's$ for which $person(x)$ is true: $\left\{\frac{x}{John}, \frac{x}{Richard}\right\}$

Quantifier

- Is the following true or false? $x > 5, x \in R$
- To make it true or false we use $\forall$ and $\exists$
  - $\forall x: [(x > 2) \rightarrow (x > 3)]\ x \in R\ (false)$

    For all real x, x>2 implies x>3.
  - $\exists x: [(x^2 = -1)]\ x \in R\ (false)$

    There exists some real x whose square is -1.

Nested Quantifiers

- Combinations of universal and existential quantification are possible:
  - $\forall x \forall y: father(x, y) \equiv \forall y \forall x: father(x, y)$
  - $\exists x \exists y: father(x, y) \equiv \exists y \exists x: father(x, y)$
  - $\forall x \exists y: father(x, y) \neq \exists y \forall x: father(x, y)$
  - $\exists x \forall y: father(x, y) \neq \forall y \exists x: father(x, y)$
  - $x, y \in \{All\ People\}$

Quiz

- Which is which?
  - $Everyone\ is\ father\ of\ someone.$
  - $Everyone\ has\ everyone\ as\ a\ father.$
  - $There\ is\ a\ person\ who\ has\ everyone\ as\ a\ father.$
  - $There\ is\ a\ person\ who\ has\ a\ father.$
  - $There\ is\ a\ person\ who\ is\ the\ father\ of\ everyone.$
  - $Everyone\ has\ a\ father.$

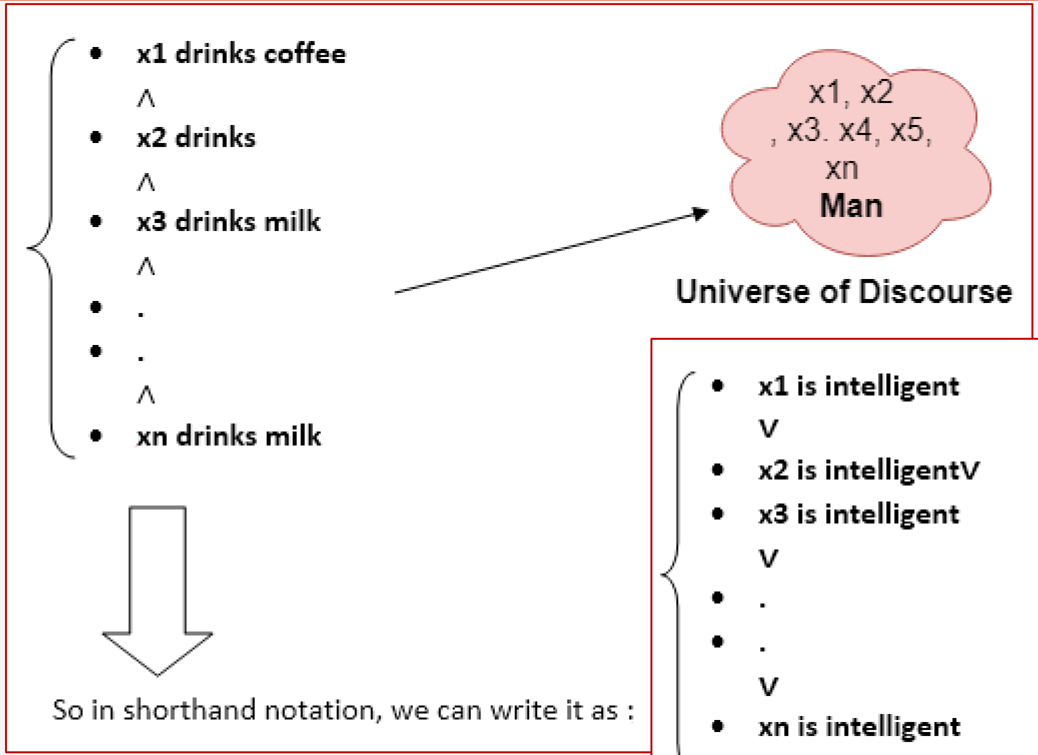
Quantifiers in FOL

Universal Quantifier (∀)

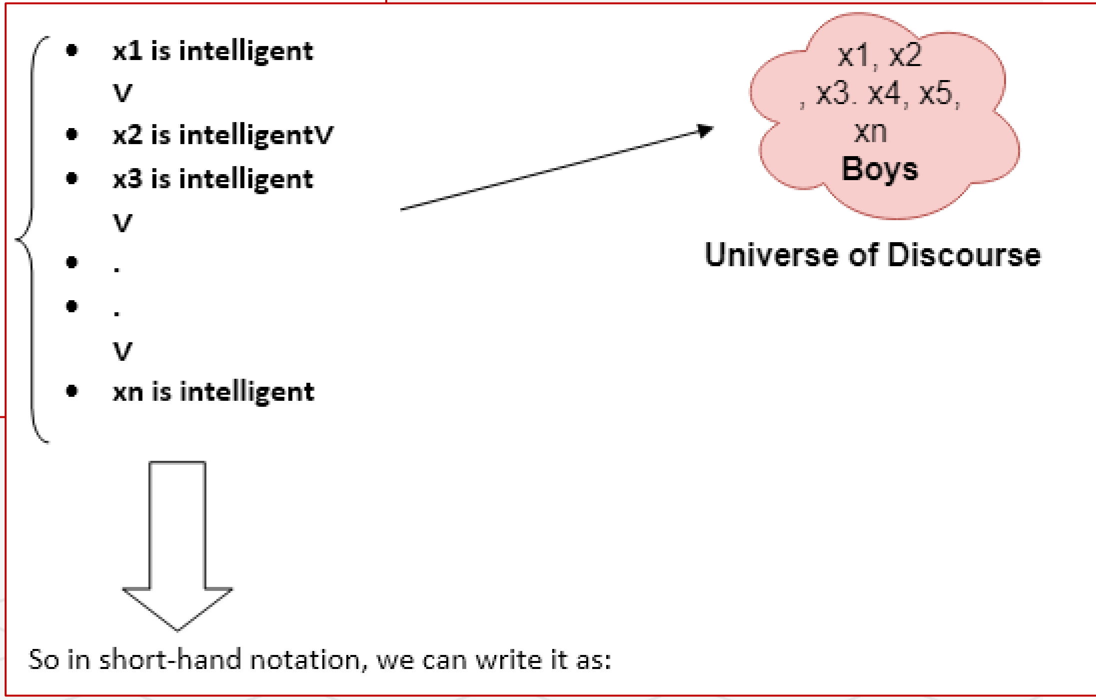
- For all x
- For each x
- For every x

Existential Quantifier (∃)

- There exists a 'x.'
- For some 'x.'
- For at least one 'x.'



$\forall x\ man(x) \rightarrow drink\ (x, coffee)$



$\exists x: boys(x) \land intelligent(x)$

Quantifier Relations

- $\forall x \forall y = \forall y \forall x$
- $\exists x \exists y = \exists y \exists x$
- $\exists x \forall y \neq \forall y \exists x$

Example#1: FOL Examples using Quantifiers

1. All birds fly.
   - $\forall_x \, bird(x) \rightarrow fly(x)$
2. Every man respects his parent.
   - $\forall_x \, man(x) \rightarrow respects(x, parent)$
3. Some boys play cricket.
   - $\exists_x \, boys(x) \rightarrow play(x, cricket)$
4. Not all students like both Mathematics and Science.
   - $\neg\forall_x \, [student(x) \rightarrow like(x, Mathematics) \wedge like(x, Science)]$
5. Only one student failed in Mathematics.
   - $\exists_x \, \left[student(x) \rightarrow failed(x, Math) \wedge \, \forall_y[\neg(x == y) \wedge \mathrm{student(y)} \rightarrow \neg\mathrm{failed(x, Math)}]\right]$

Task: Convert following set of statements to FOPL

a) Kami was a man.

b) Kami was a jhapali.

c) All jhapali were nepali.

d) Tribhuvan was a ruler.

e) All nepali were either loyal to Tribhuvan or hated him.

f) Everyone is loyal to someone.

g) People only try to assassinate rulers they are not loyal to.

h) Kami tried to assassinate Tribhuvan.

Task: Solution

a) $man(Kami)$

b) $jhapali(Kami)$

c) $\forall x: jhapali(x) \rightarrow nepali(x)$

d) $rular(Tribhuvan)$

e) $\forall x: nepali(x) \rightarrow loyal\_to(x, Tribhuvan) \vee hate(x, Tribhuvan)$

f) $\forall x \exists y: loyal\_to(x, y)$

g) $\forall x \forall y: person(x) \wedge \; ruler(y) \wedge try\_assassinate(x, y) \rightarrow \neg loyal\_to(x, y)$

h) $try\_assassinate(Kami, Tribhuvan)$

De Morgan’s Law for Quantifiers

De Morgan’s Rule

a) $P \wedge Q \equiv \sim(\sim P \vee \sim Q)$

b) $P \vee Q \equiv \sim(\sim P \wedge \sim Q)$

c) $\sim(P \wedge Q) \equiv \sim P \vee \sim Q$

d) $(P \vee Q) \equiv \sim P \wedge \sim Q$

Generalized De Morgan’s Rule

a) $\forall x: P \equiv \sim\exists x: (\sim P)$

b) $\exists x: P \equiv \sim\forall x: (\sim P)$

c) $\sim\forall x: P \equiv \exists x: (\sim P)$

d) $\sim\exists x: P \equiv \forall x: (\sim P)$

Rule is simple: if you bring a negation inside a disjunction or a conjunction, always switch between them

(or → and, and → or).

- Equality symbol: Father(John)=Henry.

This relates two objects.

$\rightarrow$ for $\forall$ and $\wedge$ for $\exists$?

Just to add a specific and rather extreme example: Consider a universe where nothing is delicious but there is at least one fruit that isn't an apple. Then that one fruit is an example showing that $\exists x: A(x) \rightarrow D(x)$ is true. (Note that this works whether or not this universe contains any apples.) – Andreas Blass Oct 19 '15 at 19:12

$A(x): x\ is\ Apple$

$D(x): x\ is\ delicious$

$\forall x: A(x) \rightarrow D(x)$ is **correct** and means all apples are delicious.

$\forall x: A(x) \wedge D(x)$ is **wrong** because this would be saying that *all fruits are apples* and *delicious* which is wrong.

Think about *which universes you want your statement to hold true in*, rather than *which values of x are "captured" by a quantifier*. In the existentially quantified statement, *surely you don't want it to hold true in a universe that doesn't contain any apples*, or one in which no apple is delicious. Similarly, in the universally quantified statement, *surely you don't want it to be false if all apples are delicious*, but *there are other fruits in the universe*. That's why it's a good idea for the implication to be vacuously true if the premise is true. – Dustan Levenstein Oct 19 '15 at 19:21

But when it comes to the existential quantifier:

$\exists x: A(x) \wedge D(x)$ is **correct**, means there is *some apple that is delicious.*

$\exists x: A(x) \rightarrow D(x)$ is **wrong**, why?. Even it says there is *some fruit* that if it is an apple, it is delicious.

There exists a fruit such that if it is an apple, then it is delicious. Let x be such a fruit.

We have two cases for what x may be here:

- $x\ is\ Apple$. Then $x\ is\ delicious$. This is the x you are searching for.
- $x\ is\ not\ Apple$. Now the statement "if x is an apple, then x is delicious" automatically holds true. Since x is not an apple, the conclusion doesn't matter. The statement is vacuously true.

So the statement $\exists x: A(x) \rightarrow D(x)$ *fails to capture precisely* your desired values of x, i.e., apples which are delicious, because it also includes other fruits.

https://math.stackexchange.com/questions/1487910/why-cant-we-use-implication-for-the-existential-quantifier

Common mistakes to avoid

- $\rightarrow$ is the main connective with $\forall$
- $\wedge$ is the main connective with $\exists$

$\forall x: king(x) \rightarrow person(x)$ $\quad x = \{Tribhuvan, Mahendra, Mukut\}$

$\forall x: king(x) \wedge person(x)$ $\quad$ All of these must be true!

$king(Tribhuvan) \wedge person(Tribhuvan)$

$king(Mahendra) \wedge person(Mahendra)$

$king(Mukut) \wedge person(Mukut)$ False!

$\exists x: king(x) \wedge person(x)$

$\exists x: king(x) \rightarrow person(x)$ $\quad$ One of these should be true!

if $king(Tribhuvan)$ $then$ $person(Tribhuvan)$

if $king(Mahendra)$ $then$ $person(Mahendra)$

if $king(Mukut)$ $then$ $person(Mukut)$ True!

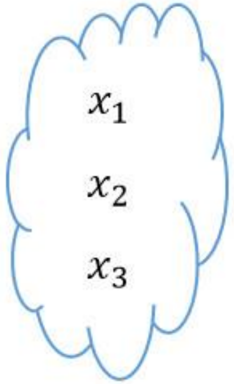
FOL Examples

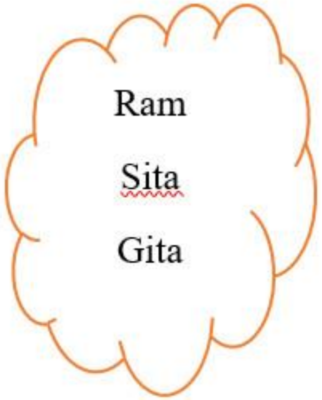
1. x is an integer.
   - $int(x)$
   - $int(a)$ – a is integer
2. Pinky is a cat.
   - $cat(pinky)$
3. Bunny is a cat.
   - $cat(Bunny)$
4. Every lion dink coffee.
   - $\forall x: lion(x) \rightarrow drink(x, Coffee)$
5. Some students are intelligent.
   - $\exists x: student(x) \wedge intelligent(x)$
6. Every student in the class has visited Pokhara or Bandipur.
   - $student(x): x\ is\ a\ student$
   - $visit\_pokhara(x): x\ has\ visited\ Pokhara$.
   - $visit\_bandipur(x): x\ has\ visited\ Bandipur$.
   - $\therefore \forall x: student(x) \rightarrow visit\_pokhara(x) \vee visit\_bandipur(x)$
7. Some prime numbers are even numbers.
   - $prime(x): x\ is\ a\ prime\ number$.
   - $even(x): x\ is\ an\ even\ number$.
   - $\therefore \exists x: prime(x) \wedge even(x)$

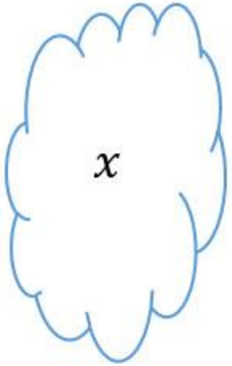
Two Variable Predicate

Two variable predicates are used in sentences like $loves, likes, wants, hates, \ldots$

1) Ram likes Sita.
   - $likes(Ram, Sita)$
2) Ram likes everyone.
   - $Ram\ likes\ x_1 \land Ram\ likes\ x_2 \land Ram\ likes\ x_3$
   - $likes(Ram, x_1) \land likes(Ram, x_2) \land likes(Ram, x_3)$
   - $i.e.\ \forall x: likes(Ram, x)$
3) Everyone likes everyone.







$Ram\ likes\ everyone. \land Sita\ likes\ everyone. \land Gita\ likes\ everyone.$

$$\left.\begin{array}{l}\forall x: likes(Ram, x)\\ \forall x: likes(Sita, x)\\ \forall x: likes(Gita, x)\end{array}\right\} \boldsymbol{\forall y \forall x: likes(y, x)}$$

Two Variable Predicate (2)

4) Someone likes everyone.
   - Hari likes everyone.
   - $\forall x: likes(Hari, x)$
   - $\exists y \forall x: likes(y, x)$

$$Hari\ likes\ x_1. \lor Hari\ likes\ x_2. \lor Hari\ likes\ x_3.$$

5) Someone likes someone.
   - $\exists y \exists x: likes(y, x)$
   - $\exists x \exists y: likes(x, y)$
6) Everyone likes someone.
   - $\forall x \exists y: likes(x, y)$
7) Everyone is liked by someone.
   - Someone likes everyone: $\exists \boldsymbol{x} \forall y: likes(x, y)$
   - $\exists x \forall y: liked\_by(y, x)$ or $\forall x \exists y: liked\_by(x, y)$
8) Ram is liked by someone.
   - $\exists x: liked\_by(Ram, x)$ or $\exists x: likes(x, Ram)$
9) Someone is liked by everyone.
   - $\exists x \forall y: liked\_by(x, y)$ or $\exists x \forall y: likes(y, x)$
10) Nobody likes everyone.
   - Ram does not like everyone. $\neg \forall x: likes(Ram, x)$
   - $\therefore \forall \boldsymbol{y}(\neg \forall \boldsymbol{x}: \boldsymbol{likes}(\boldsymbol{y}, \boldsymbol{x}))$

Facts into Predicates

Convert the following statements in predicate logic:-

Sub: x predicate: p

① Ram is tall. $p(x)$

$\text{tall}(\text{Ram})$

② Ram loves Sita.

$\text{loves}(\text{Ram}, \text{Sita})$

③ Ram teaches either math or C.

$\text{teaches}(\text{Ram}, \text{math}) \vee \text{teaches}(\text{Ram}, C)$

④ Ram teaches math if and only if Ram does not teach C.

$\text{teaches}(\text{Ram}, \text{math}) \leftrightarrow \neg\, \text{teaches}(\text{Ram}, C)$ ~/¬

① All students like football.

$\forall x: \text{student}(x) \rightarrow \text{like}(x, \text{football})$

② Some students like football.

$\exists x: \text{student}(x) \wedge \text{like}(x, \text{football})$

③ All students are happy.

$\forall x: \text{student}(x) \rightarrow \text{happy}(x)$

④ Some students are happy.

$\exists x: \text{student}(x) \wedge \text{happy}(x)$

① Every rational numbers are real numbers

$\forall x: \text{rational-number}(x) \rightarrow \text{real-number}(x)$

ran(x): x is rational no.

ren(x): x is real no.

$\forall x: \text{ran}(x) \rightarrow \text{ren}(x)$

② Some real numbers are rational numbers

$\exists x: \text{ren}(x) \wedge \text{ran}(x)$

③ Not every real number is a rational number.

$\sim \forall x: \text{ren}(x) \rightarrow \text{ran}(x)$

Class Work

1. Marcus was a man.
2. Marcus was a Pompeian.
3. All Pompeian’s were roman.
4. Every gardener likes sun.
5. All purple mushrooms are poisonous.
6. Everyone loves everyone.
7. Everyone is loyal to everyone.
8. Everyone is loyal to someone.
9. Everyone loves everyone except himself.
10. All romans were either loyal to Ceasor or hated him.

Class Work (2)

11. People only try to assassinate rulers they are not loyal to.
12. Anyone who is married and has more than one spouse is a Bigamist.
13. You can fool all the people some of the time.
14. You can fool some of the people all time.
15. Every city has a dog catcher who has been bitten by dog in the town.
16. One’s husband is one’s male spouse.
17. Parents and child are inverse relation.
18. A sibling is another child of one’s parent.
19. The best score in AI is always higher that the best score in NM.
20. Prem is a barber who shares all men in the city who does not shave himself.

Task

Represent the following facts in predicate logic:

a) Snow is white.

b) If it rains then sky will be cloudy.

c) If you will not work hard, you will fail.

d) All indoor games are easy.

e) Sandip only likes cricket games.

f) All dogs are mammal.

g) Roses are red.

h) Dasarath is father of Ram.

Task

Represent the following in symbol logic:

a) All that glitters is not gold.

b) Any person who is respected by every person is a king.

c) God helps those who help themselves.

d) Admand & Tenjing went up the hill.

e) Ram likes Sita.

Task with Solution

Express following sentences in FOPL:

1. Ram is a boy. $\Rightarrow boy(Ram)$
2. Rita is mother of Ram. $\Rightarrow mother(Rita, Ram)$
3. Fido is a dog. $\Rightarrow dog(Fido)$
4. Apple is food.
   - $\Rightarrow food(Apple)$
5. All animals will die.
   - $\Rightarrow \forall x\ [animal(x) \rightarrow die(x)]$
6. Dogs are animal.
   - $\Rightarrow \forall x: dog(x) \rightarrow animal(x)$
7. No mango is blue.
   - $\Rightarrow blue(Mango)$
8. Every girls like ice cream.
   - $\Rightarrow \forall x: girl(x) \rightarrow like(x, Ice\ Cream)$
9. Some boys like monkey.
   - $\Rightarrow \exists x: boy(x) \land like(x, Monkey)$
10. John like all kind of food.
    - $\Rightarrow \forall x: food(x) \rightarrow like(John, x)$
11. Everybody is loyal to someone.
    - $\Rightarrow \forall x: dog(x) \rightarrow animal(x)$
12. All cat have tail and whisker.
    - $\Rightarrow \forall x: cat(x) \rightarrow have\_tail(x) \land have\_whisker(x)$
13. Steve only like easy course.
    - $\Rightarrow like(Steve, Easy\ Course)$
14. Science course are hard.
    - $\Rightarrow \forall x: course(x) \rightarrow hard(Science)$

Proof by Reduction

Example#1:

a) Fido is a dog.
b) All dogs are animal.
c) All animals will die.
d) Prove: Fido will die.

STEP 1: Facts into FOL

a) $dog(Fido)$
b) $\forall x: dog(x) \rightarrow animal(x)$
c) $\forall x: animal(x) \rightarrow die(x)$
d) $Conclusion: die(Fido)$

STEP 2: Proof by Reduction or Tabulation Method

SN	Statements	Reason
1	$dog(Fido)$	Given
2	$\forall x: dog(x) \rightarrow animal(x)$	Given
3	$\forall x: animal(x) \rightarrow die(x)$	Given
4	$dog(Fido) \rightarrow animal(Fido)$	Universal Instantiation
5	$animal(Fido) \rightarrow die(Fido)$	Universal Instantiation
6	$animal(Fido)$	Applying Modus Ponens in 1 & 4
7	$die(Fido)$	Applying Modus Ponens in 5 & 6

Task: Proof by Reduction

Example#2:

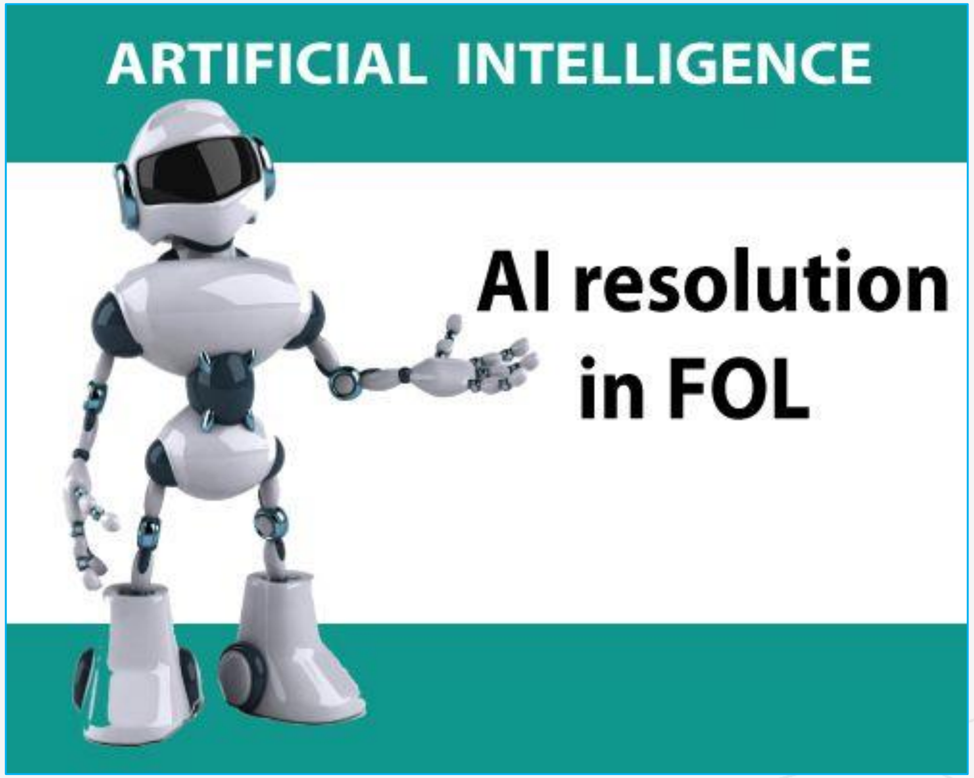
a) All people who are not poor & are smart are happy.
b) Those people who reads are not stupid.
c) John can read & is wealth.
d) Happy people have exciting life.
e) Prove: John has exciting life.

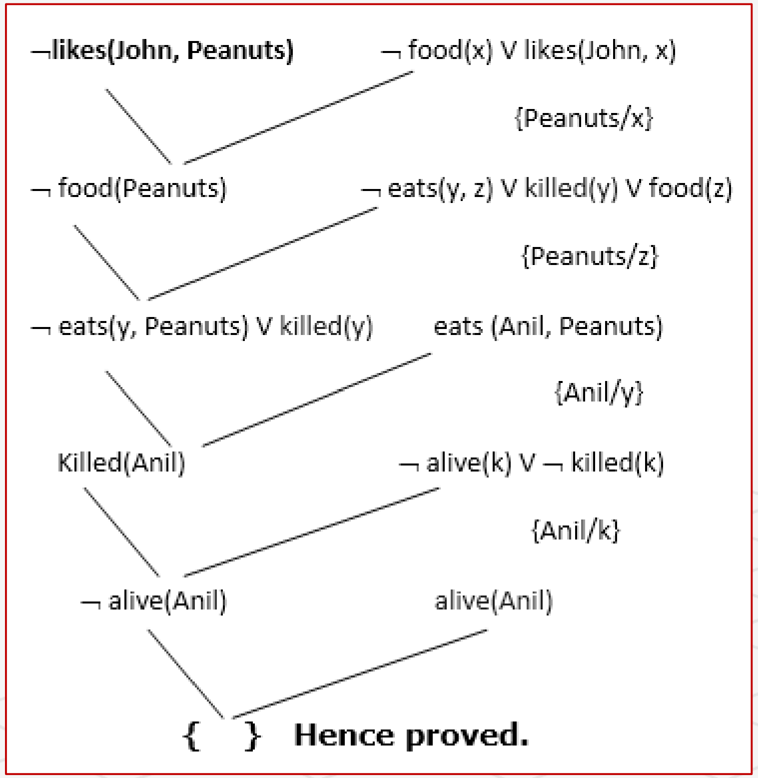
STEP 1: Facts into FOL

e) $\forall x[\sim poor(x) \wedge smart(x)] \rightarrow happy(x)$
f) $\forall x: reads(x) \rightarrow smart(x)$
g) $read(John) \wedge \sim poor(John)$
h) $\forall x: happy(x) \rightarrow excitingLife(x)$
i) $Conclusion: excitingLife(John)$

4.3. Rules of Inference, Unification, Resolution Refutation System (RRS), Answer Extraction from RRS, Rule Based Deduction System

Rules of Inference, Unification, Resolution Refutation System (RRS), Answer Extraction from RRS, Rule Based Deduction System





Unification

- Unification is a process of making two different logical atomic expressions identical by finding a substitution.
- Unification depends on the substitution process.
- It takes two literals as input and makes them identical using substitution.

For Example:

$p(x, f(y))$ $p(2, f(g(z)))$

$[2/x, g(z)/y] \leftarrow Substitution\ Set$

2 for x, $g(z)$ for y

Substitution: A substitution t_i/v_i specifies substitution of term t_i to a variable v_i.

Resolution Refutation System (RSS)

Resolution in FOL

- It is a *theorem proving technique* that *proves by contradiction*.
- It is used if there are various statements are given and we need to prove a conclusion of those statements.
- ***Unification*** is a key concept in proof by resolution.
- Resolution is a single inference rule which efficiently operates on CNF or causal form.

Conjunctive Normal Form (CNF)

- A sentence represented as a conjunction of clauses is said to be CNF.
- Clause: Disjunction of literals is called clause.

Steps for Resolution

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

Resolution Example#1

Example#1:

a) It is sunny and warm, you will enjoy.
b) It is raining, you will get wet.
c) It is a warm day.
d) It is raining.
e) It is sunny.
f) **Prove: You will enjoy.**

STEP 1: Facts into FOL Conversion

a) $sunny \wedge warm \rightarrow enjoy$
b) $raining \rightarrow wet$
c) $warm$
d) $raining$
e) $sunny$
f) enjoy

STEP 2: FOL into CNF

1. **Eliminate → & rewrite**
   a. $\sim[\boldsymbol{sunny} \wedge \boldsymbol{warm}] \vee \boldsymbol{enjoy}$ $[Since\ a \rightarrow b = \sim a \vee b]$
   b. $\sim raining \vee wet$
   c. $warm$
   d. $raining$
   e. $sunny$
   f. Enjoy
2. **Move ~ inwards & rewrite**
   a. $\boldsymbol{\sim sunny \wedge \sim warm} \vee enjoy$
   b. $\sim raining \vee wet$
   c. $warm$
   d. $raining$
   e. $sunny$
   f. enjoy
3. **Rename or Standardize Variables**
   a. No variables used, so sentences remain same.
4. **Remove ∃ by elimination**
   a. Since there is no ∃, so sentences remain same.
5. **Drop ∀ -** Since there is no ∀, so sentences remain same.

STEP 3: Negate the statement to be proved

- $\sim enjoy$

STEP 4: Draw resolution graph

$\sim enjoy$ $\boldsymbol{\sim sunny \wedge \sim warm} \vee enjoy$

$\boldsymbol{\sim sunny \wedge \sim warm}$ $sunny$

$\boldsymbol{\sim warm}$ $warm$

{ } **Hence proved.**

Resolution Example#2

Example#2:

a. John likes all kind of food.
b. Apple and vegetable are food
c. Anything anyone eats and not killed is food.
d. Anil eats peanuts and still alive
e. Harry eats everything that Anil eats.
f. **Prove: John likes peanut.**

Solution: -

STEP 1: Facts into FOL Conversion

a) $\forall x: food(x) \rightarrow likes(John, x)$
b) $food(Apple) \wedge food(vegetables)$
c) $\forall x \forall y: eats(x, y) \wedge \sim killed(x) \rightarrow food(y)$
d) $eats(Anil, Peanuts) \wedge alive(Anil)$
e) $\forall x: eats(Anil, x) \rightarrow eats(Harry, x)$
f) $\forall x: \sim killed(x) \rightarrow alive(x)$
g) $\forall x: alive(x) \rightarrow \sim killed(x)$
h) $likes(John, Peanuts)$

STEP 2: FOL into CNF

1. **Eliminate → & rewrite**
   a. $\forall x: \sim food(x) \vee likes(John, x)$ $[Since\ a \rightarrow b = \sim a \vee b]$
   b. $food(Apple) \wedge food(vegetables)$
   c. $\forall x \forall y: \sim[eats(x, y) \wedge \sim killed(x)] \vee food(y)$
   d. $eats(Anil, Peanuts) \wedge alive(Anil)$
   e. $\forall x: \sim eats(Anil, x) \vee eats(Harry, x)$
   f. $\forall x: \sim[\sim killed(x)] \vee alive(x)$
   g. $\forall x: \sim alive(x) \vee \sim killed(x)$
   h. $likes(John, Peanuts)$
2. **Move ~ inwards & rewrite**
   a. $\forall x: \sim food(x) \vee likes(John, x)$
   b. $food(Apple) \wedge food(vegetables)$
   c. $\forall x \forall y: \sim eats(x, y) \vee killed(x) \vee food(y)$ [De-Morgan's Law $\neg(P \wedge Q) = \neg P \vee \neg Q$]
   d. $eats(Anil, Peanuts) \wedge alive(Anil)$
   e. $\forall x: \sim eats(Anil, x) \vee eats(Harry, x)$
   f. $\forall x: killed(x) \vee alive(x)$
   g. $\forall x: \sim alive(x) \vee \sim killed(x)$
   h. $likes(John, Peanuts)$

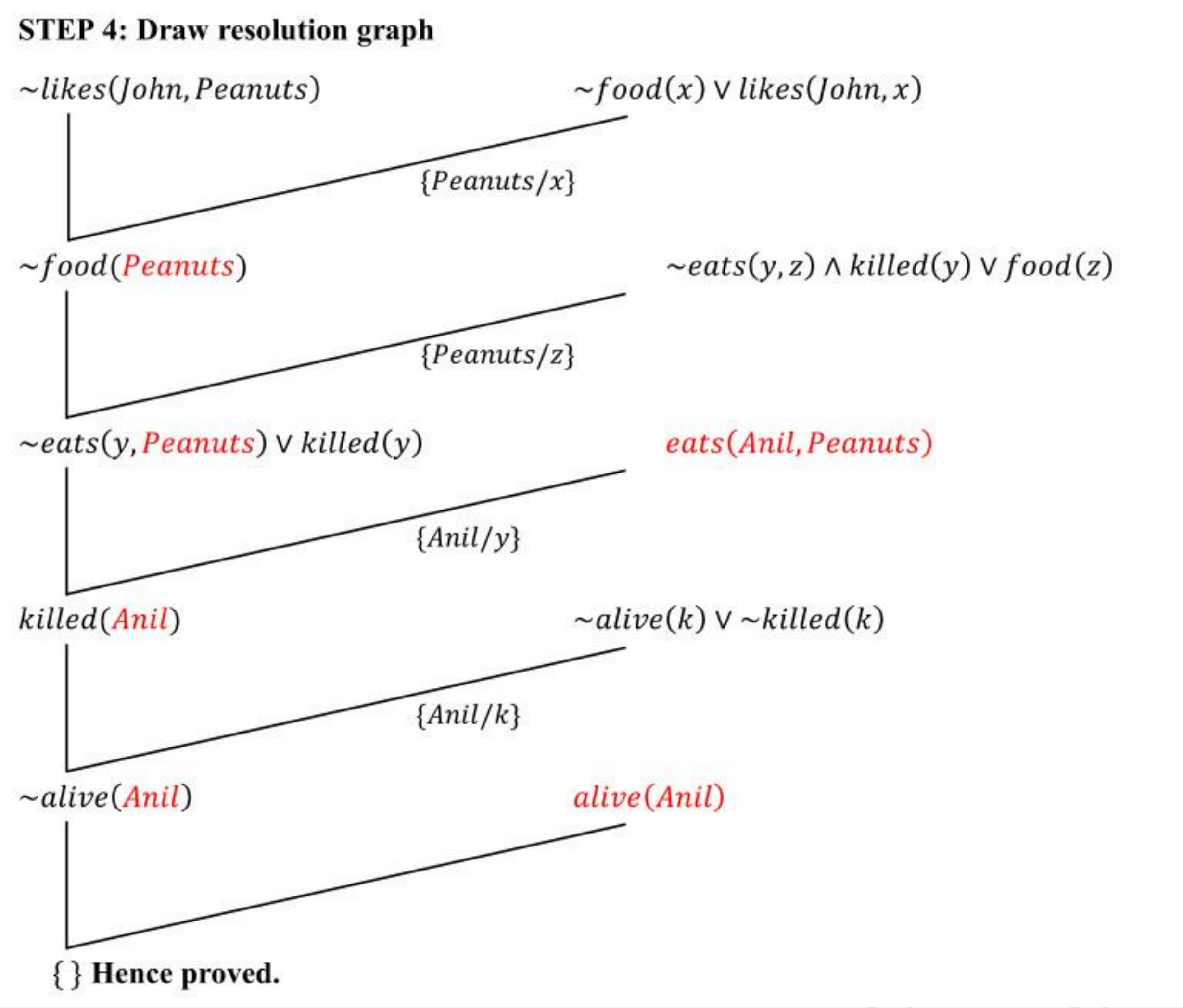
Resolution Example#2 (2)

3. **Rename or Standardize Variables**
   a. $\forall x: \sim food(x) \vee likes(John, x)$
   b. $food(Apple) \wedge food(vegetables)$
   c. $\forall y \forall x: \sim eats(y, z) \vee killed(y) \vee food(z)$
   d. $eats(Anil, Peanuts) \wedge alive(Anil)$
   e. $\forall w: \sim eats(Anil, w) \vee eats(Harry, w)$
   f. $\forall g: killed(g) \vee alive(g)$
   g. $\forall k: \sim alive(k) \vee \sim killed(k)$
   h. $likes(John, Peanuts)$
4. **Skolemization - Remove ∃ by elimination**
   a. Since there is no ∃, so sentences remain same.
5. **Drop ∀**
   a. $\sim food(x) \vee likes(John, x)$
   b. $food(Apple) \wedge food(vegetables)$
   c. $\sim eats(y, z) \vee killed(y) \vee food(z)$
   d. $eats(Anil, Peanuts) \wedge alive(Anil)$
   e. $\sim eats(Anil, w) \vee eats(Harry, w)$
   f. $killed(g) \vee alive(g)$
   g. $\sim alive(k) \vee \sim killed(k)$
   h. $likes(John, Peanuts)$
6. **Distribute conjunction ∧ over disjunction**
   a. $\sim food(x) \vee likes(John, x)$
   b. $food(Apple)$
   c. $food(vegetables)$
   d. $\sim eats(y, z) \vee killed(y) \vee food(z)$
   e. $eats(Anil, Peanuts)$
   f. $alive(Anil)$
   g. $\sim eats(Anil, w) \vee eats(Harry, w)$
   h. $killed(g) \vee alive(g)$
   i. $\sim alive(k) \vee \sim killed(k)$
   j. $likes(John, Peanuts)$

STEP 3: Negate the statement to be proved

- $\sim likes(John, Peanuts)$

Resolution Example#2 (3)



Hence the negation of the conclusion has been proved as a complete contradiction with the given set of statements.

Explanation of Resolution Graph:

- In 1st step of resolution graph, ¬likes(John, Peanuts), & likes(John, x) get resolved(canceled) by substitution of {Peanuts/x}, & we are left with ¬food(Peanuts).
- In 2nd step of the resolution graph, ¬food(Peanuts), and food(z) get resolved (canceled) by substitution of { Peanuts/z}, & we are left with ¬eats(y, Peanuts) v killed(y) .
- In 3rd step of the resolution graph, ¬eats(y, Peanuts) & eats(Anil, Peanuts) get resolved by substitution {Anil/y}, & we are left with Killed(Anil).
- In 4th step of the resolution graph, Killed(Anil) and ¬killed(k) get resolve by substitution {Anil/k}, & we are left with ¬alive(Anil).
- In the last step of the resolution graph, ¬alive(Anil) and alive(Anil) get resolved.

Class Work

Proof by Resolution Refutation

a) Fido is a dog.

b) All dogs are animal.

c) All animals will die.

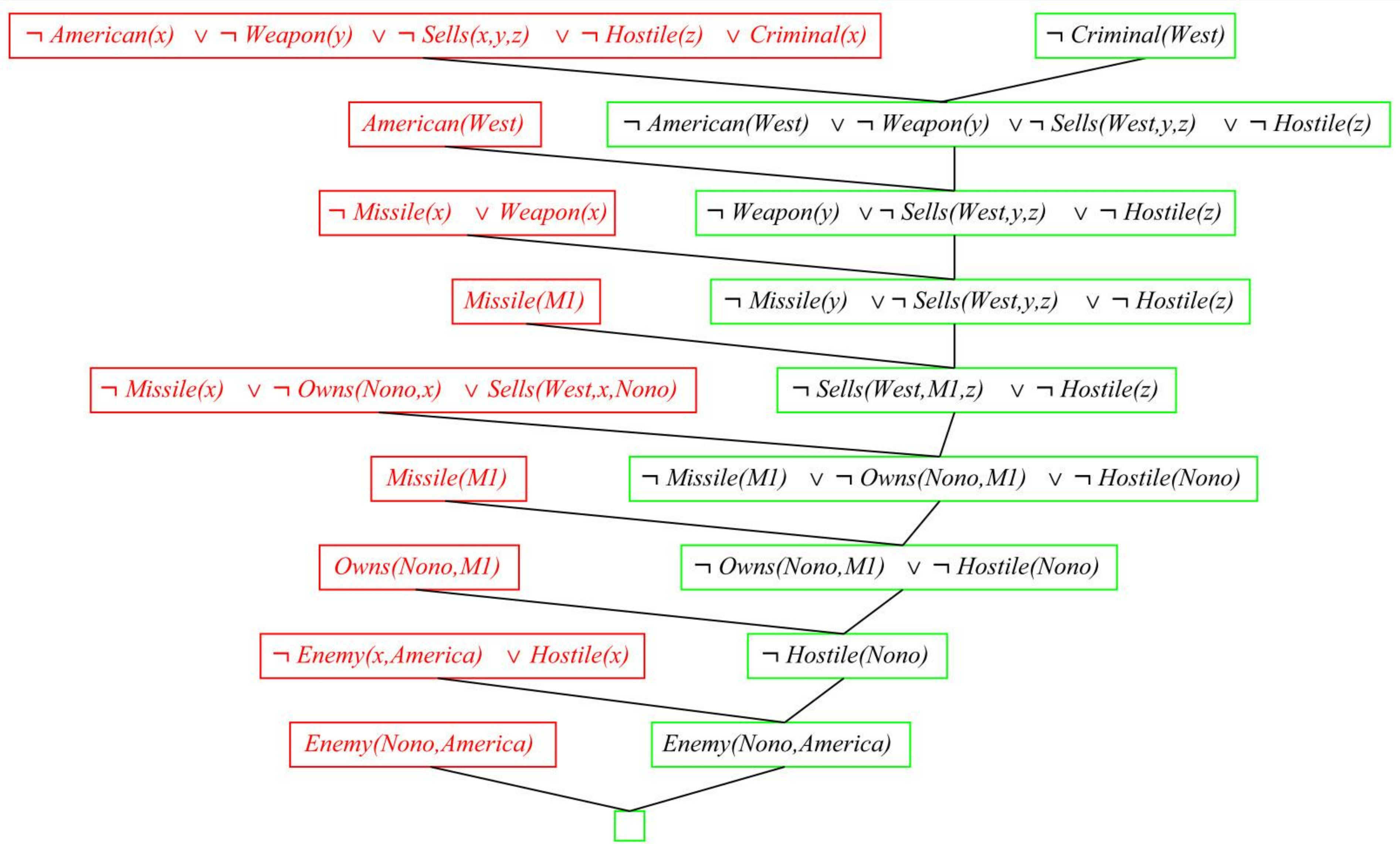
d) Prove: Fido will die.

Tasks (Submit only two)

1. Given premises "Every American who sells weapons to hostile nations is a criminal. The country XYZ is enemy of America. All of its missiles in XYZ were sold by Donald, who is American". *Prove that Donald is a criminal* by using FOPL based resolution.
2. Using resolution, solve the following statements: All Pompeian are Romans. All Romans were either loyal to Caesar or hated him. Everyone is loyal to someone. People only try to assassinate rulers they not loyal to. Marcus tried to assassinate Caesar. Marcus was Pompeian. *Find, did Marcus assassinate Caesar?*
3. Assume the following facts:
   - Horses, cows, pigs are mammals
   - An offspring of a horse is a horse
   - Bulebeard is a horse
   - Blebeard is Charlie's parent
   - Offspring and parent are inverse relations
   - Every mammal has a parent

   Prove Charlie is a horse using resolution refutation.

Unification Graph#1



Unification Graph#2

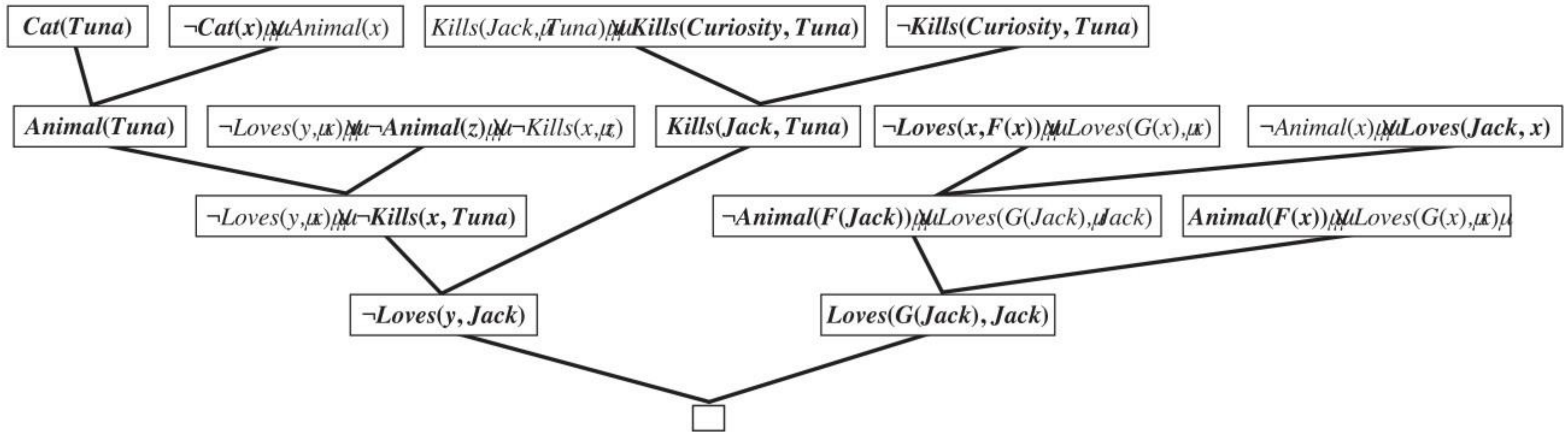
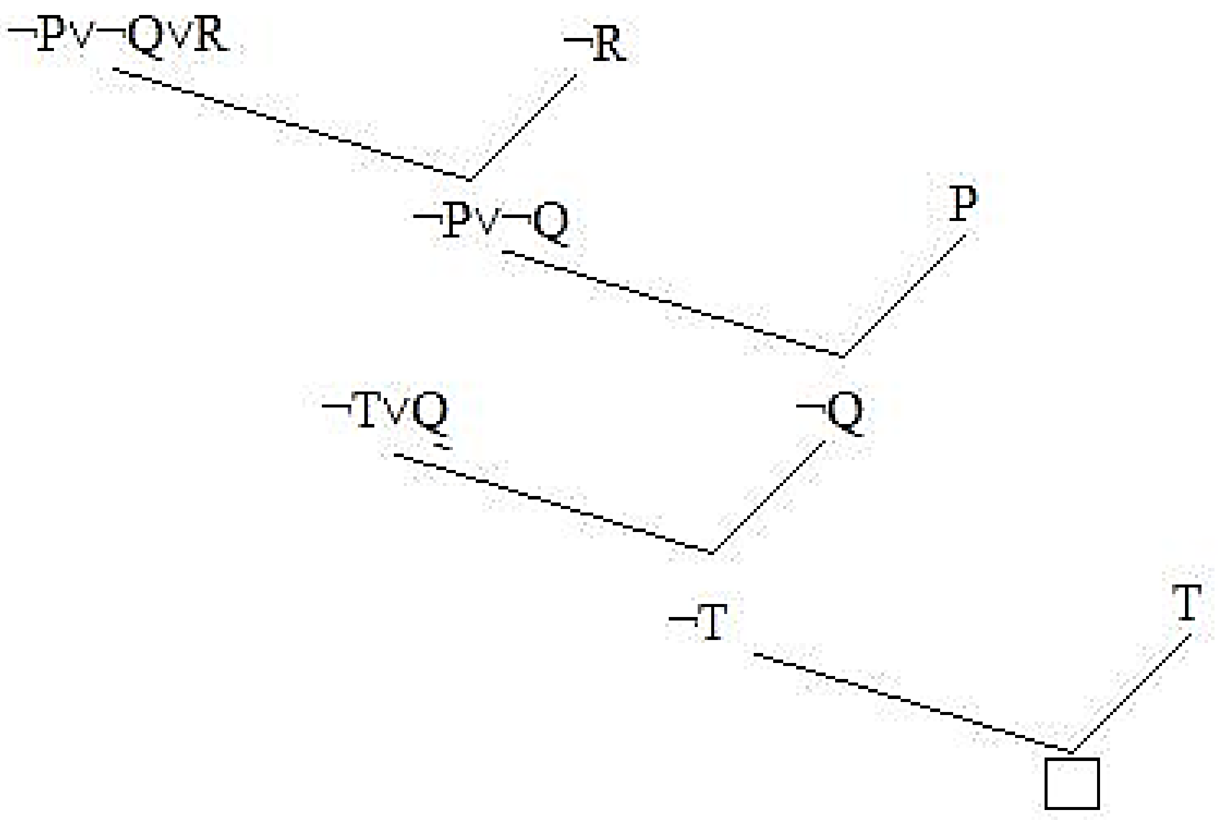


Figure 9.12 A resolution proof that Curiosity killed the cat. Notice the use of factoring in the derivation of the clause $Loves(G(Jack), Jack)$. Notice also in the upper right, the unification of $Loves(x, F(x))$ and $Loves(Jack, x)$ can only succeed after the variables have been standardized apart.

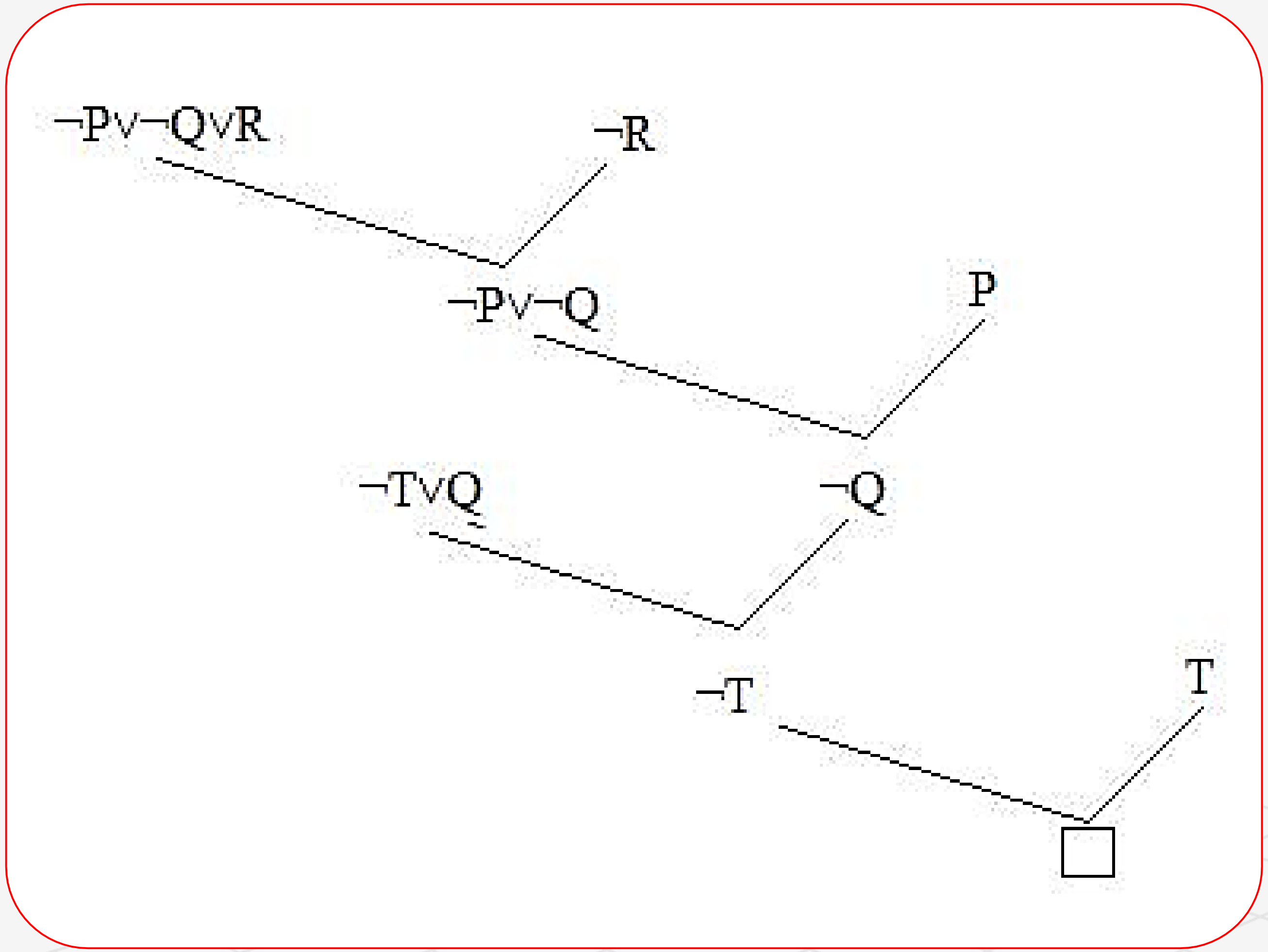
Answer Extraction from RRS

- *Negate the answer* that should extract
- If the result becomes $\{\,\}$ or $\emptyset$ or $NILL$ the conclusion proved by default otherwise conclusion failed.



Answer Extraction from RRS (2)

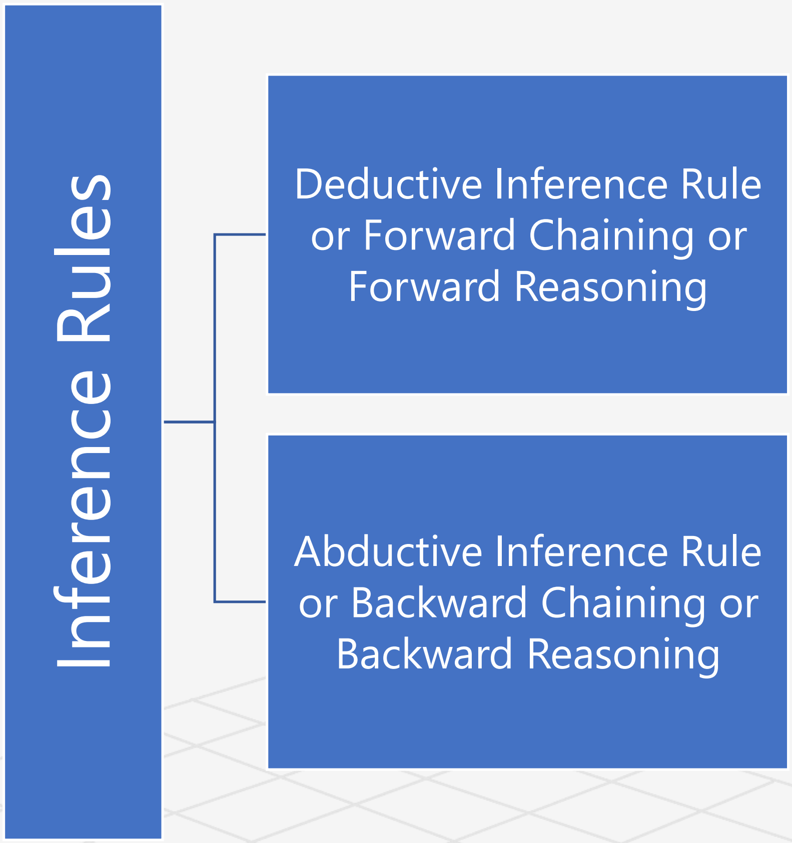
- *Negate the answer* that should extract
- If the result becomes $\{\,\}$ or $\emptyset$ or $NILL$ the conclusion proved by default otherwise conclusion failed.



Rule Based Deduction System

Inference Engine:

- An *Inference Engine* is a tool of *Artificial Intelligence* that is used as a component of the system to deduce new information from a knowledge base using logical rules and reasoning. The first-ever Inference Engines were a part of expert systems in AI.
- It predicts outcomes with the already existing pool of data, comprehensively analyzing it and using logical reasoning to predict the outcomes.
- IE *works in one of the two ways*, either *data-driven* or *goal-driven*, which later came to be known as ***forwarding chaining*** and ***backward chaining***.

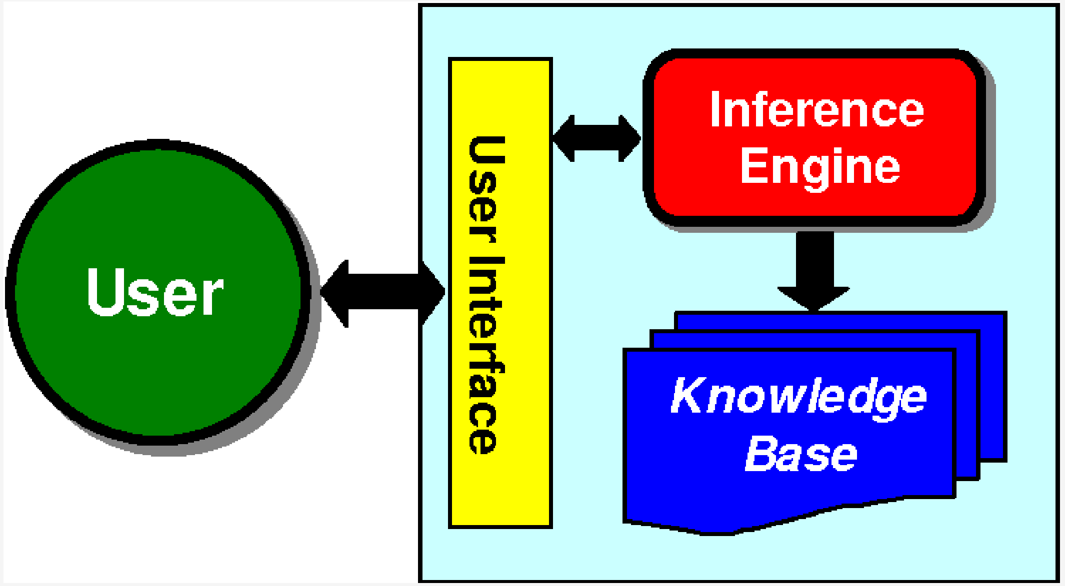


Inference: Forward & Backward Chaining

Horn Form

- **Horn Clause:** At most one literal is Positive
- For Example
  - $(\sim P \vee \sim Q \vee V)$ is a Horn Clause
  - $\sim P \vee \sim W$ is a Horn Clause
  - $\sim P \vee Q \vee V$ is not a Horn Clause
- **Definite Clauses:** exactly one literal is positive.
- Horn clauses can be re-written as implications
  - Proposition symbol (fact) or
  - Conjunction of symbols (body or premise) → symbol (head)
  - Example: $(\sim C \vee \sim B \vee A)$ becomes $C \wedge B \rightarrow A$

Rule Based Inference Engine



Inference Engine

- It is component of system that applies logical rules to the KB to deduce new information.
- It works on two modes:
    1. Forward Chaining/Reasoning
    2. Backward Chaining/Reasoning

Forward Chaining

- Starts with the known facts and asserts new facts

Backward Chaining

- Starts with goals and works backward to determine what facts must be asserted so that the goals can be achieved.

Deductive Inference Rule or Forward Chaining

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

A

$\underline{A \rightarrow B}$

B

Example:

$It\ is\ raining.$

$If\ it\ is\ raining, the\ street\ is\ wet.$

$The\ street\ is\ wet.$

Abductive Inference Rule or Backward Chaining

Backward Chaining: Conclude from “B” and “A implies B” to “A”.

B

$\underline{A \rightarrow B}$

A

Example:

$The\ street\ is\ wet.$

$If\ it\ is\ raining, the\ street\ is\ wet.$

$It\ is\ raining.$

Knowledge Base Example

Prove: The Donald is a criminal.

a) It is crime for an American to sell weapons to the enemy of America.

b) China is an enemy of America.

c) China has some missiles.

d) All the missiles were sold to China by Donald.

e) Missile is a weapon.

f) Donald is American.

Solution:- *Facts to FOL*

a) $american(x) \wedge weapon(y) \wedge sell(x, y, z) \wedge enemy(z, America) \rightarrow criminal(x)$

b) $\boldsymbol{enemy(China, America)}$

c) $\exists x: missile(x) \wedge has(China, x)$

d) $\forall x: missile(x) \wedge \boldsymbol{has(China, x)} \rightarrow sell(Donald, x, China)$

e) $\boldsymbol{missile(x)} \rightarrow weapon(x)$

f) $\boldsymbol{american(Donald)}$

Forward Chaining Proof (Data Driven)

STEP 1: Take non-repeating LHS part.

$american(Donald)$

$missile(x)$

$has(China, x)$

$enemy(China, America)$

Forward Chaining Proof (2)

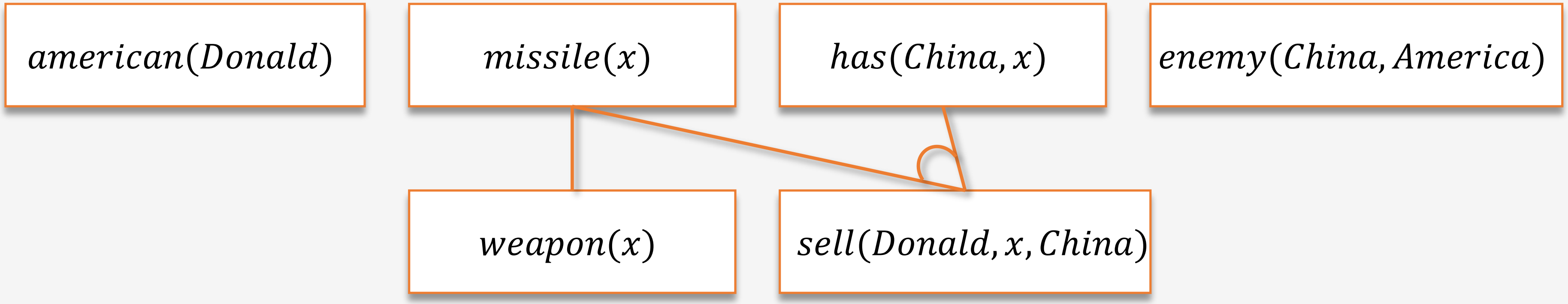
STEP 2: If RHS part comes to root note then write stepwise. e) $\boldsymbol{missile(x)} \rightarrow weapon(x)$

$american(Donald)$ | $missile(x)$ | $has(China, x)$ | $enemy(China, America)$

$$
\begin{array}{c}
missile(x) \\
| \\
weapon(x)
\end{array}
$$

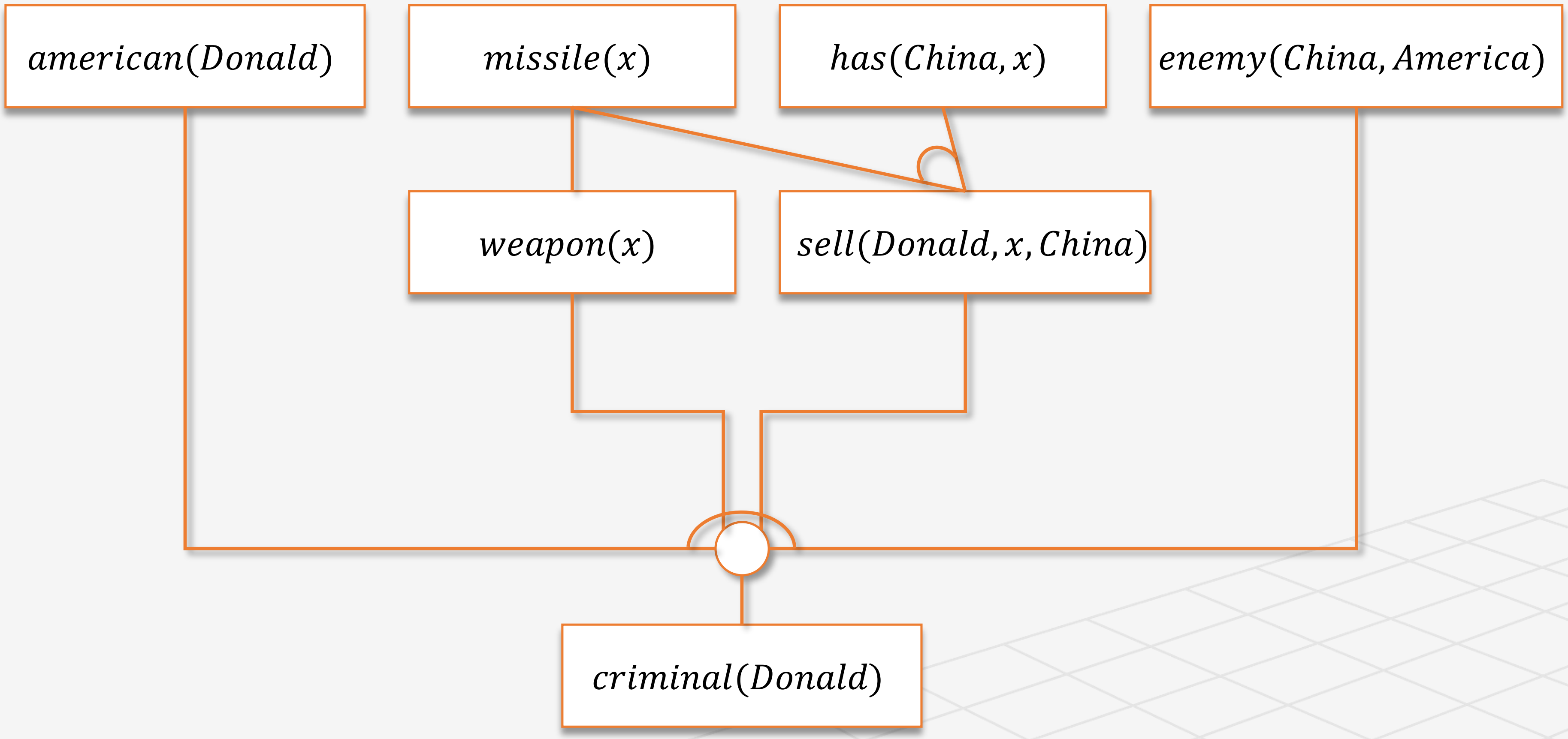
Forward Chaining Proof (3)

STEP 3: If RHS part comes to root note then write stepwise. d) $\forall x: missile(x) \wedge \boldsymbol{has(China, x)} \rightarrow sell(Donald, x, China)$



Forward Chaining Proof (3)

STEP 4: Substitution {Donald/x}. a) $american(x) \wedge weapon(y) \wedge sell(x,y,z) \wedge enemy(z, America) \rightarrow criminal(x)$



Backward Chaining (Decision/Goal Driven)

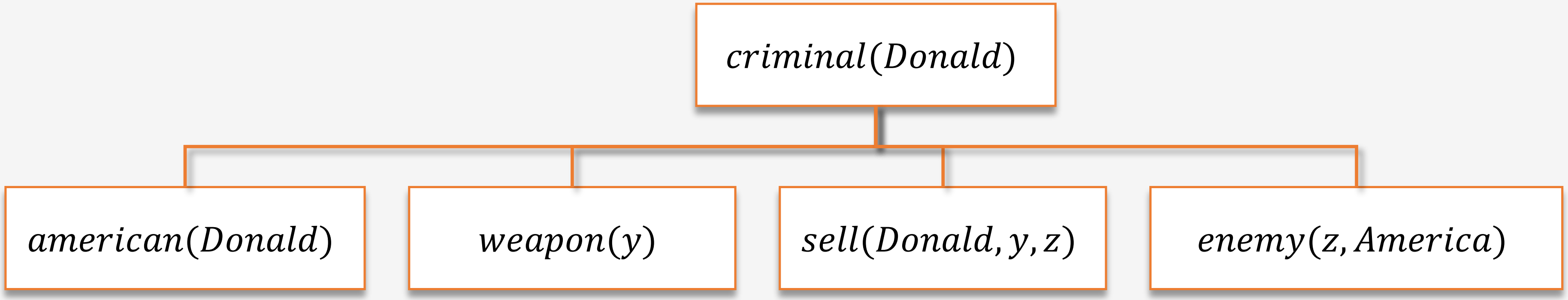
STEP 1: goal state to initial state $criminal(Donald)$

$criminal(Donald)$

Backward Chaining (2)

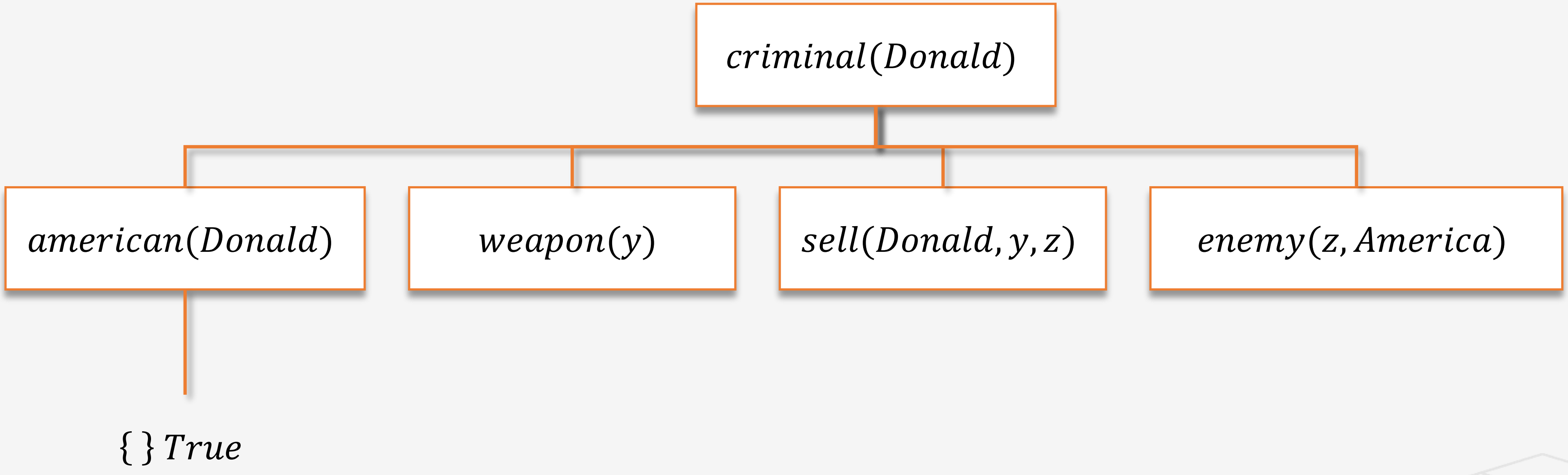
STEP 2: Find the RHS part if the root node is connected.

$$american(x) \land weapon(y) \land sell(x, y, z) \land enemy(z, America) \rightarrow \boldsymbol{criminal(x)}$$



Backward Chaining (3)

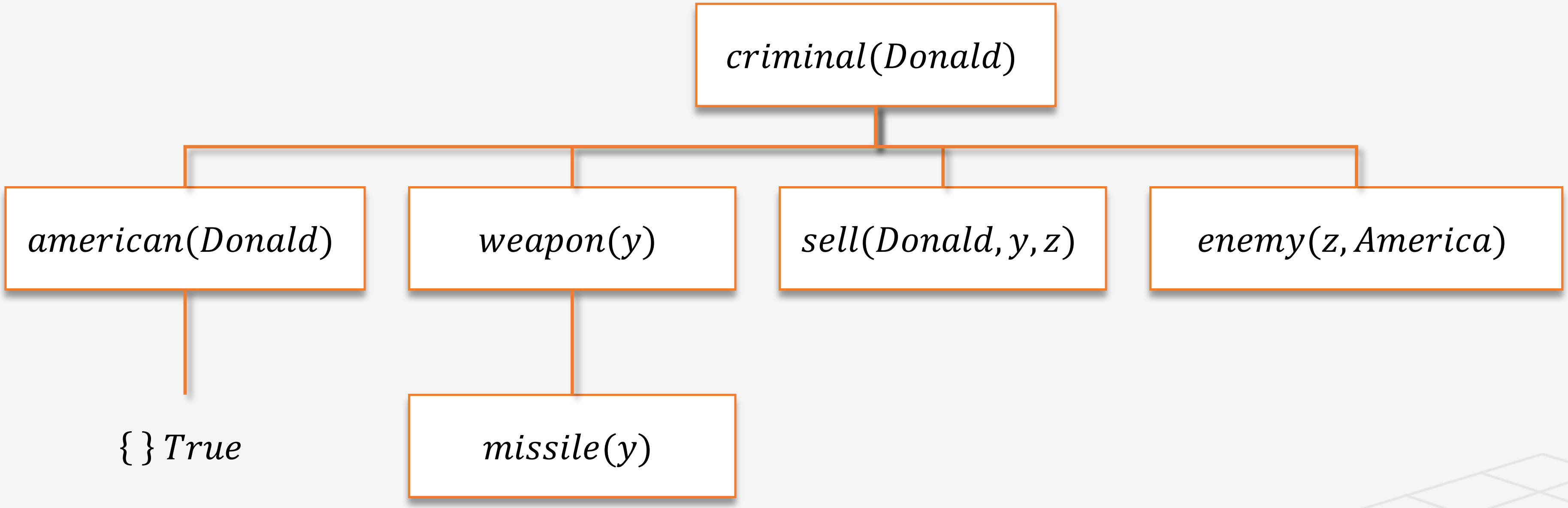
STEP 3: If $american(Donald)$ is in RHS Part? If not then we assume true. *Directly present in FOL.*



Backward Chaining (4)

STEP 4: e) $\boldsymbol{missile(x)} \rightarrow weapon(x)$

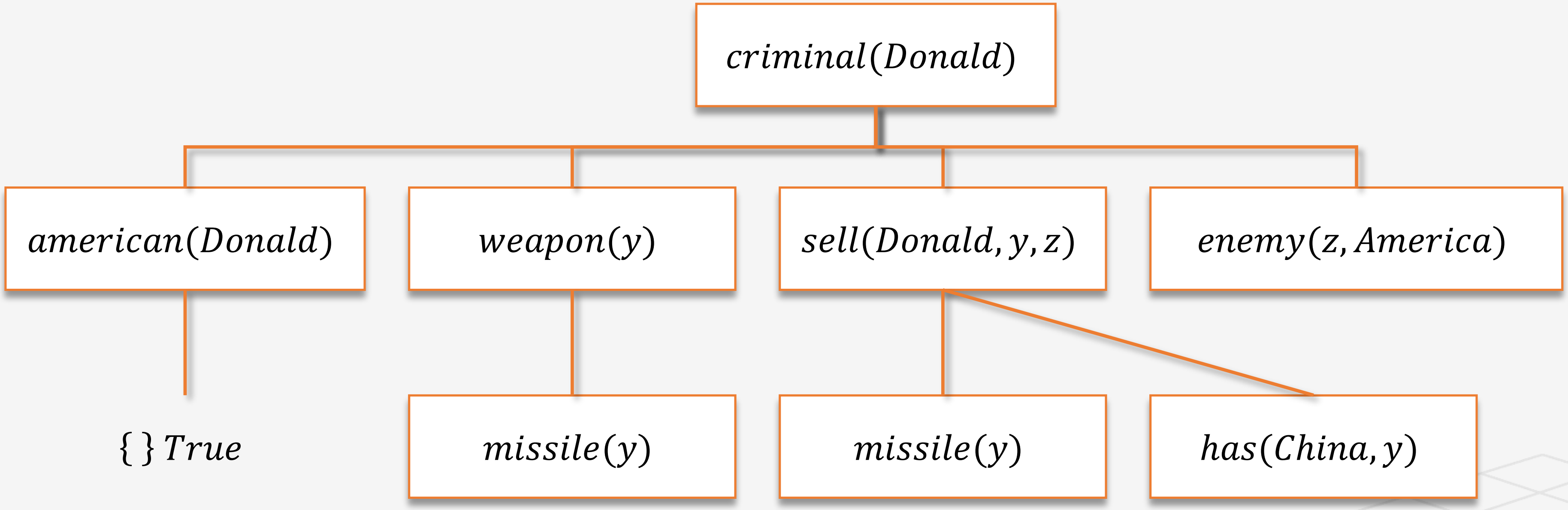
$\boldsymbol{missile(y)}$ Can be derived from $weapon(y)$.



Backward Chaining (5)

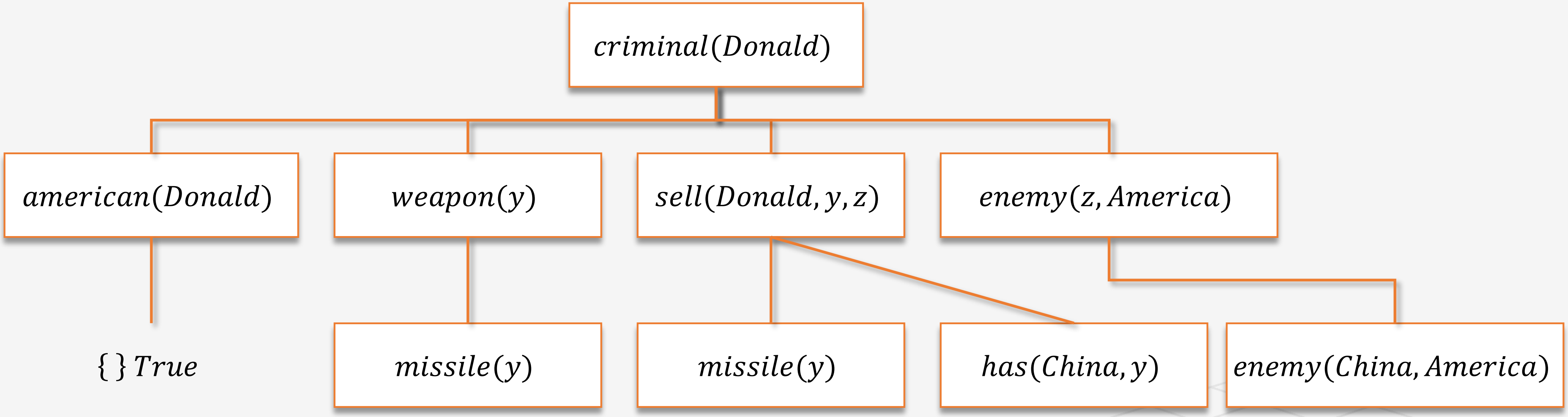
STEP 5: *d)* $\forall x\!: missile(x) \wedge \boldsymbol{has(China, x)} \rightarrow sell(Donald, x, China)$

$missile(y)$, $\boldsymbol{has(China, y)}$ can be derived from $sell(Donald, y, z)$.



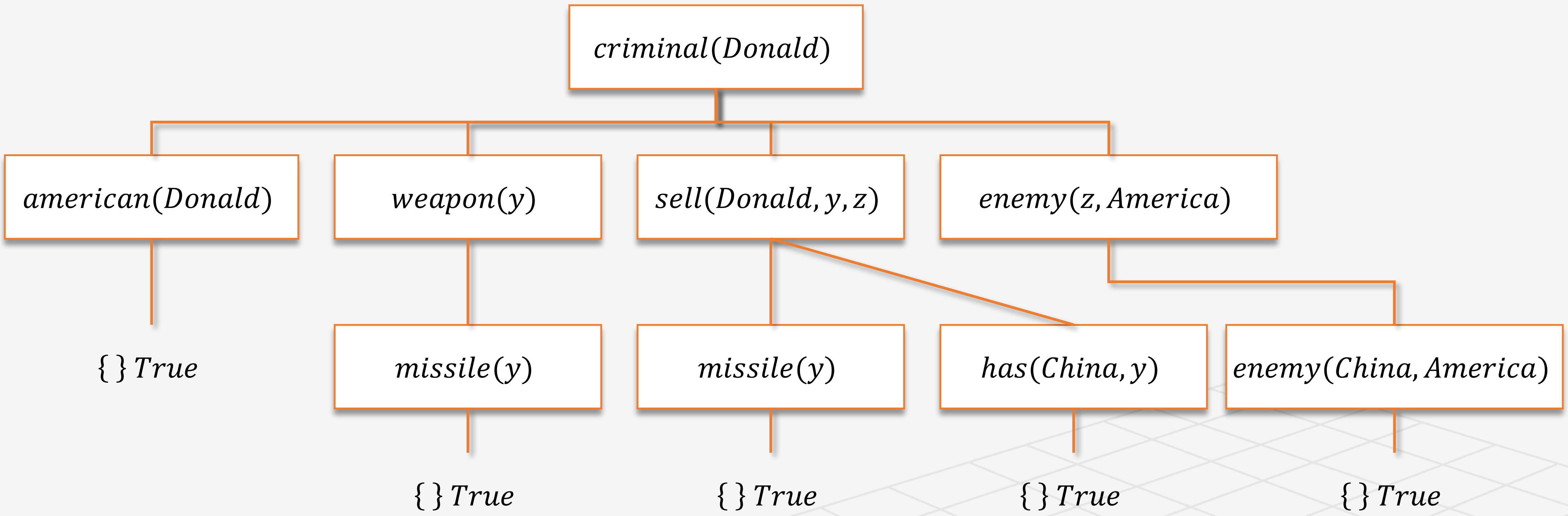
Backward Chaining (6)

STEP 6: b) $enemy(China, America)$



Backward Chaining (7)

STEP 7: Remaining are FOL, so all are true.



4.4. Statistical Reasoning - Probability and Bayes' Theorem and Causal Networks, Reasoning in Belief Network

Statistical Reasoning - Probability and Bayes' Theorem and Causal Networks, Reasoning in Belief Network

Statistical Reasoning

- There are several techniques that can be *used to augment* **knowledge representation** techniques with statistical measures that describe levels of evidence and belief.
- An important goal for many problem solving systems is to collect evidence as the systems goes along and to modify its behavior, we need a statistical theory of evidence.
- Bayesian statistics is such a theory which stresses the conditional probability as fundamental notion.

Probability: Descriptions of the likelihood of some event occurring (ranging from 0 to 1).

Bayes' Theorem

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

Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \Rightarrow P(A|B).P(B) = P(A \cap B)$$

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \Rightarrow P(B|A).P(A) = P(B \cap A)$$

From RHS:

$$P(A \cap B) = P(A|B).P(B) = P(B|A).P(A)$$

A = Hypothesis; B = Evidence or Data

P(A/B) – Probability of hypothesis given that we have given the data.

P(B/A) – Probability of data given that the hypothesis is true.

P(A) – Probability of hypothesis before observing the evidence.

P(B) – Pure/Marginal Probability of data.

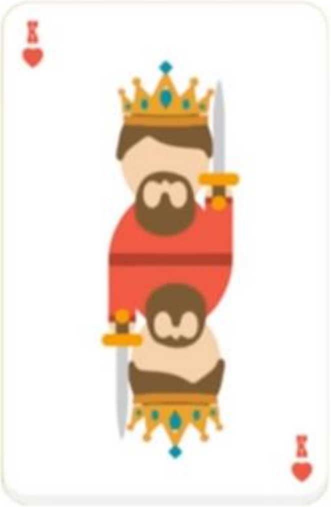
Example: Find the probability of being King, you have given a face card.

$$P(King|Face) = \frac{P(Face|King).P(King)}{P(Face)} = \frac{1 * 4/52}{12/52} = \frac{1}{3}$$

Bayes' Theorem (2)

Given a hypothesis H and evidence E , Bayes' theorem states that the relationship between the probability of the hypothesis before getting the evidence P(H) and the probability of the hypothesis after getting the evidence P(H|E) is

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$



$$P(King) = 4/52 = 1/13$$

$$P(King|Face) = \frac{P(Face|King).P(King)}{P(Face)}$$

$$= \frac{1.(1/13)}{3/13} = 1/3$$

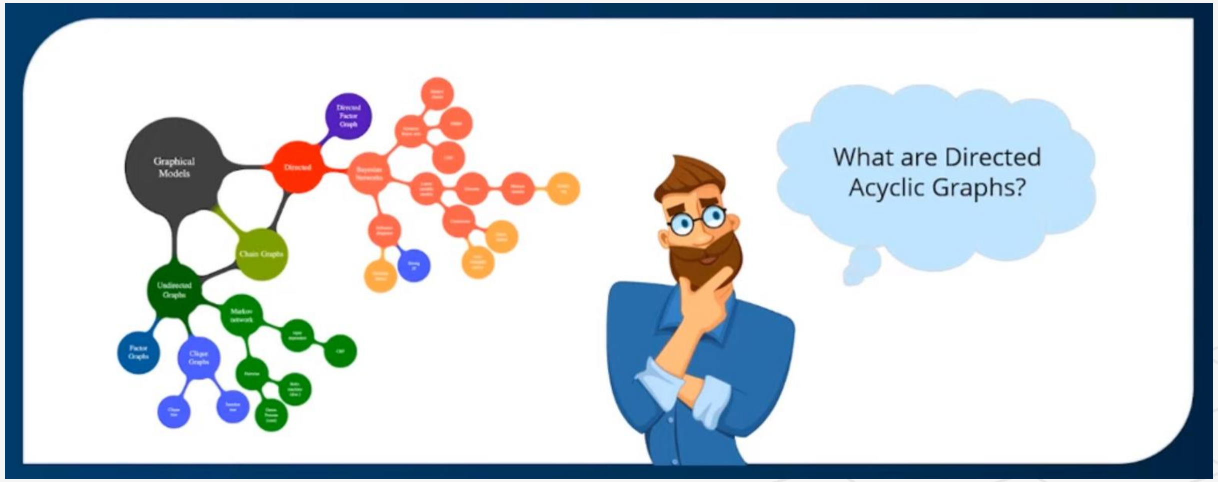
$$P(Face|King) = 1$$

$$P(Face) = 12/52 = 3/13$$

Causal Bayesian Networks (CBN)

- Bayesian Network – A Bayesian Network falls under the category of Probabilistic Graphical Modeing (PGM) technique that is used to compute uncertainties by using the concept of probability.

- CBN: A flexible tool to enable fairer machine learning
  - CBN as a Visual Tool - Characterizing patterns of unfairness underlying a dataset
  - CBN as a Quantitative Tool - Path-specific (counterfactual) inference techniques for fairness

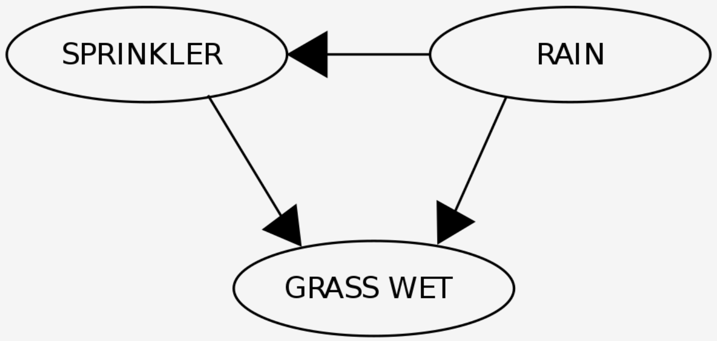


Bayesian/Causal Networks

- A causal network is a Bayesian network with the requirement that the relationships be causal.
- A Bayesian network (also known as a Bayes network, belief network, or decision network) is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Bayesian networks are ideal for taking an event that occurred and predicting the likelihood that any one of several possible known causes was the contributing factor.

- Wikipedia, 2020

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Fig: A simple Bayesian network with conditional probability tables

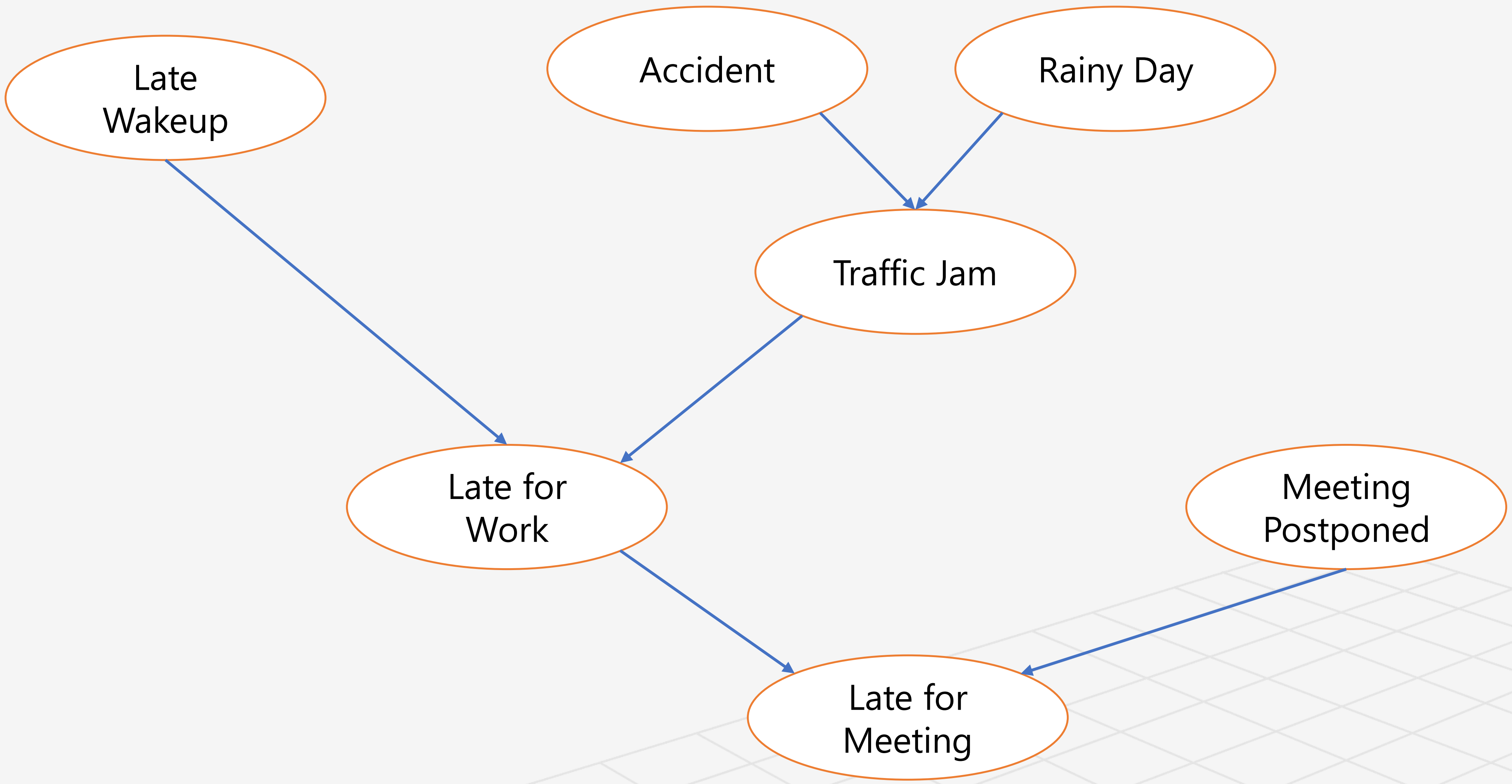
Reasoning in Belief Network

Probabilistic Reasoning

- It is a statistical method of reasoning based on some numeric value assigned to them.
- It involves four methods:
  1. **Bayesian Belief Networks**
  2. Dempster Shaffer Theory
  3. Reasoning with Certainty Factor
  4. Fuzzy Reasoning

Inference is about taking in data and trying to draw a conclusion based on the limited information you have in a situation where not all the data is given.
Reasoning is about drawing conclusions using your own mind.

Representation in Bayesian Belief Networks



How Naïve Bayes Classifier related to Bayesian Network?

A **Bayesian network** is a graphical model that represents a set of variables and their conditional dependencies.

For example, disease and symptoms are connected using a network diagram. All symptoms connected to a disease are used to calculate the probability of the existence of the disease.

Naive Bayes classifier is a technique to assign class labels to the samples from the available set of labels. This method assumes each feature's value as independent and will not consider any correlation or relationship between the features.

Example: Naïve Bayes Classifier

Question: Which fruit is yellow, sweet & long among the given data?

$Fruit = \{Yellow, Sweet, Long\}$

Fruit	Yellow	Sweet	Long	*Total*
Orange	350	450	0	650
Banana	400	300	350	400
Others	50	100	50	150
Total	*800*	*850*	*400*	***1200***

Remember, Bayes' Theorem: $P(A|B) = \frac{P(B|A).P(A)}{P(B)}$

Numerical Solution: Naïve Bayes Classifier

For Orange:

$$P(Yellow|Orange) = \frac{P(Orange|Yellow).P(Yellow)}{P(Orange)} = \frac{\frac{350}{800} * \frac{800}{1200}}{650/1200} = 0.5385$$

$$P(Sweet|Orange) = \cdots = 0.69$$

$$P(Long|Orange) = \cdots = 0$$

$$\therefore \boldsymbol{P(Fruit|Orange) = 0.5385 * 0.69 * 0 = 0}$$

For Banana:

$$P(Yellow|Banana) = \cdots =?$$

$$P(Sweet|Banana) = \cdots =?$$

$$P(Long|Banana) = \cdots =?$$

$$\therefore \boldsymbol{P(Fruit|Banana) = \cdots = 0.65}$$

For Other:

$$P(Yellow|Other) = \cdots =?$$

$$P(Sweet|Other) = \cdots =?$$

$$P(Long|Other) = \cdots =?$$

$$\therefore \boldsymbol{P(Fruit|Other) = \cdots = 0.072}$$

Maximum probability show that the required fruit is Banana.

Questions

1. Given premises "Every American who sells weapons to hostile nations is a criminal. The country XYZ is enemy of America. All of its missiles in XYZ were sold by Donald, who is American". Prove that Donald is a criminal by using FOPL based resolution refutation method. [8] (75/1, 69/5)

2. Assume the following facts: [2+6] (74/5, 72/6)
   - John likes all kinds of food.
   - Apples are food.
   - Chicken is food.
   - Anything anyone eats and isn't killed by is food.
   - Bill eats peanuts and is still alive.
   - Sue eats everything bill eats.

   Prove that John likes peanuts using resolution.

3. Using resolution, solve: All Pompeian are Romans. All Romans were either loyal to Caesar or hated him. Everyone is loyal to someone. People only try to assassinate rulers they not loyal to. Marcus tried to assassinate Caesar. Marcus was Pompeian. Find, did Marcus assassinate Caesar? [7] (72/10)

Questions (2)

1. Assume the following facts: [8] (73/5)
    - Horses, cows, pigs are mammals
    - An offspring of a horse is a horse
    - Bulebeard is a horse
    - Blebeard is Charlie's parent
    - Offspring and parent are inverse relations
    - Every mammal has a parent

    Prove Charlie is a horse using resolution refutation.

2. Why do we need FOPL? State any three rules of inference. How can we make the machine with learning capacity? [2+3+3] (70/10)

3. All over smart persons are stupid. Children of over smart persons are naughty. Ram is children of Hari. Hari is over smart. Show that Ram is naughty using FOPL based resolution method. [8] (70/5)

Links to Further Reading

1. What is Knowledge Representation in AI? Techniques You Need To Know – edureka!
2. Knowledge Representation – JavaTpoint
3. Knowledge Representation & NLP in AI
4. What is a Knowledge Base?
5. https://deepmind.com/blog/article/Causal_Bayesian_Networks
6. http://www.brainkart.com/article/Bayesian-Networks-and-Certainty-Factors_8590/
7. https://www.youtube.com/watch?v=Fv_LGQKgWi0
8. http://www.lirmm.fr/~mugnier/ArticlesPostscript/FullTR-RR2012KonigLeclereMugnierThomazoV2.pdf

Q/A?

Artificial Intelligence

Thank You!

Er. Shiva K. Shrestha

Head, Dept. of Computer Engineering

Khwopa College of Engineering

ai.khwopa@gmail.com | computer.khwopa@gmail.com

https://github.com/ErSKS/AI