4. Knowledge Representation, Inference and Reasoning

Knowledge:

→ Knowledge is a theoretical or practical understanding of a subject or a domain and it is also the sum of what is currently known. Hence, knowledge is the sum of what is known: the body of truth, information, and principles acquired by mankind.

→ Knowledge according to Sunasee and Sewery, 2002

"Knowledge is human proficiency stored in a person's mind, gained through experience, and interaction with the person's environment."

→ In general, knowledge is more than just data, it consist of: facts, ideas, beliefs, heuristics, associations, rules, abstractions, relationships, customs.

→ Research literature classifies knowledge as follows:

- Classification-based Knowledge » Ability to classify information
- Decision-oriented Knowledge » Choosing the best option
- Descriptive knowledge » State of some world (heuristic)
- Procedural knowledge » How to do something
- Reasoning knowledge » What conclusion is valid in what situation?
- Assimilative knowledge » What its impact is?

Knowledge Representation

Knowledge representation (KR) is the study of how knowledge about the world can be represented and what kinds of reasoning can be done with that knowledge. Knowledge Representation is the method used to encode knowledge in Intelligent Systems.

Some issues that arise in knowledge representation from an AI perspective are:

- How do people represent knowledge?
- What is the nature of knowledge and how do we represent it?
- Should a representation scheme deal with a particular domain or should it be general purpose?
- How expressive is a representation scheme or formal language?
- Should the scheme be declarative or procedural?

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

→ Representational Adequacy

  the ability to represent the required knowledge;

→ Inferential Adequacy

  the ability to manipulate the knowledge represented to produce new knowledge corresponding to that inferred from the original;

→ Inferential Efficiency

  the ability to direct the inferential mechanisms into the most productive directions by storing appropriate guides;

→ Acquisitional Efficiency

  the ability to acquire new knowledge using automatic methods wherever possible rather than reliance on human intervention.

An **axiom** is a sentence or proposition that is not proved or demonstrated and is considered as self-evident or as an initial necessary consensus for a theory building or acceptation. According as requirements, the new sentences are added to the knowledge base and then new sentences are also derived from old axiom & theorems, called **inference**.

Logic is method of reasoning process in which conclusions are drawn from premises using rules of inference. The logic is knowledge representation technique that involves:

- **Syntax:** defines well-formed sentences or legal expression in the language
- **Semantics:** defines the "meaning" of sentences
- **Inference rules:** for manipulating sentences in the language

Basically, the logic can be classified as:

- Proposition (or statements or calculus) logic
- Predicate [or First Order Predicate Logic (FOPL)] logic

A. Propositional Logic

A proposition is a declarative sentence to which only one of the “Truth value” (i.e. TRUE orFALSE) can be assigned (but not both). Hence, the propositional logic is also called Booleanlogic. When a proposition is true, we say that its truth value is T, otherwise its truth value is F.

For example:

- The square of 4 is 16 →T
- The square of 5 is 27 →F

The sentences of propositional logic can be categories as: atomic sentences and complex sentences.

- **Atomic Sentences**(Simple)

  The atomic sentences consist of a single proposition symbol. Each such symbol stands for aproposition that can be true or false. We use symbols that start with an uppercase letter andmay contain other letters or subscripts, for example: p, q, r, s etc.

  For example:

  p = Sun rises in West. (False sentence)

- **Complex Sentences (**molecular or combined or compound)

  The two or more statements connected together with some logical connectives suchas AND (∧), OR (∨), Implication (→), etc. There are five connectives in common use:

Name	**Representation**	**Meaning**
Negation	$\neg p$	not p
Conjunction (true when both statement are true, otherwise false)	$p \wedge q$	p and q
Disjunction (false when both statement are false, otherwise true)	$p \vee q$	p or q (or both)
Exclusive Or (false when both statement are same)	$p \oplus q$	either p or q, but not both
Implication (false when p is true and q is false)	$p \rightarrow q$	if p then q
Bi-conditional or Bi-implication (true when both statement have same truth value)	$p \leftrightarrow q$	p if and only if q
The order of precedence in propositional logic is (from highest to lowest): Inverse, AND, OR, Implication and Double Implication.		

Truth Table

p	**q**	**$\neg p$**	**$p \wedge q$**	**$p \vee q$**	**$p \oplus q$**	**$p \rightarrow q$**	**$p \leftrightarrow q$**
T	T	F	T	T	F	T	T
T	F	F	F	T	T	F	F
F	T	T	F	T	T	T	F
F	F	T	F	F	F	T	T

Converse:	If $p \rightarrow q$ is an implication, then its converse is $q \rightarrow p$
Inverse:	If $p \rightarrow q$ is an implication, then its inverse is $\neg p \rightarrow \neg q$
Contrapositive:	If $p \rightarrow q$ is an implication, then its contrapositive is $\neg q \rightarrow \neg p$

Q.Verify that $p \leftrightarrow q$ is equivalent to $(p \rightarrow q) \wedge (q \rightarrow p)$

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

Q.Construct the truth table of $\neg(p \wedge q) \vee (r \wedge \neg p)$

p	q	r	$p \wedge q$	$\neg(p \wedge q)$	$\neg p$	$r \wedge \neg p$	$\neg(p \wedge q) \vee (r \wedge \neg p)$
T	T	T	T	F	F	F	F
T	T	F	T	F	F	F	F
T	F	T	F	T	F	F	T
T	F	F	F	T	F	F	T
F	T	T	F	T	T	T	T
F	T	F	F	T	T	F	T
F	F	T	F	T	T	T	T
F	F	F	F	T	T	F	T

Q.

Logical equivalence	Two proposition p and q are logically equivalent and written as if both p and q have identical truth values
	Eg: $\neg (p \wedge q) \equiv (\neg p \vee \neg q)$ (hint: draw the truth value for proof) ⇒The following logical equivalences apply to any statements; the p's, q's and r's can stand for atomic statements or compound statements. i. Double Negative Law $\neg(\neg p) \equiv p$ ii. De Morgan's Laws $\neg(p \vee q) \equiv (\neg p) \vee (\neg q)$ $\neg(p \wedge q) \equiv (\neg p) \vee (\neg q)$ iii. Distributive Laws $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
Tautology	If a proposition have a truth value for every interpretation
	E.g.: $p \vee \neg p$ (hint: draw the truth value for proof)
Contradiction	If a proposition have a false value for every interpretation
	E.g.: $p \wedge \neg p$ (hint: draw the truth value for proof)
Contingent	If a proposition have both true and false value
	E.g.: $p \wedge q$ (hint: draw the truth value for proof)

Q. Write converse, inverse, negation, implication, contrapositive of the following integration.

"The program is well structured only if it is readable"

Soln:

Here,

p = "the program is readable"

q = "the program is well structured"

a) Implication: $(p \rightarrow q)$

If the program is readable, then it is well structured.

b) Converse: $(q \rightarrow p)$

If the program is well structured, then it is readable.

c) Inverse: $(\neg p \rightarrow \neg q)$

If the program is not readable, then it is not well structured.

d) Contrapositive: $(\neg q \rightarrow \neg p)$

If the program is not well structured, then it is not readable

e) Negation of p: $(\neg p)$

The program is not readable.

Q. There are two restaurants next to each other. One has a sign board as: "Good food is not cheap". The other has a sign board as "Cheap food is not good". Are both the sign board saying the same thing?

Soln:

Here, let's assume:

G= "Food is Good"

C= "Food is Cheap"

Now, Sentence 1:- "Good food is not cheap" can be symbolically written as $G \rightarrow \neg C$

Similarly,

Sentence 2:- "Cheap food is not good" can be symbolically written as $C \rightarrow \neg G$

Now, The Truth Table is:

G	C	$\neg G$	$\neg C$	$G \rightarrow \neg C$	$C \rightarrow \neg G$
T	T	F	F	F	F
T	F	F	T	T	T
F	T	T	F	T	T
F	F	T	T	T	T

Since, $G \rightarrow \neg C$ and $C \rightarrow \neg G$ are logically equivalent. So both are saying same thing.

Rules of Inference

The process of drawing conclusion from given premises in an argument is called inference. To draw the conclusion from the given statements, we must be able to apply some well-defined steps that helps reaching the conclusion.

The inference algorithm is sound if *everything returned is a needle* (hence some needles may be missed) and complete if *all needles are returned* (hence some hay may be returned too).

Let S be the set of all right answers.

A **sound** algorithm never includes a wrong answer in S, but it might miss a few right answers. => not necessarily "complete".

A **complete** algorithm should get every right answer in S: include the complete set of right answers. But it might include a few wrong answers. It might return a wrong answer for a single input. => not necessarily "sound".

The steps of reaching the conclusion are provided by the rules of inference.

a. Modus Ponens Rule

$$\begin{array}{l} p \rightarrow q \\ \underline{p\qquad} \\ \therefore q \end{array}$$

Example:

- If Ram is hard working, then he is intelligent.
- Ram is hard working.

∴ Ram is intelligent.

b. Modus Tollens Rule

$$\begin{array}{l} p \rightarrow q \\ \underline{\neg q\qquad} \\ \therefore \neg p \end{array}$$

Example:

- We will go swimming only if it is sunny.
- It is not sunny.

∴ We will not go swimming.

c. Hypothetical Syllogism Rule

$$\begin{array}{l} p \rightarrow q \\ \underline{q \rightarrow r\qquad} \\ \therefore p \rightarrow r \end{array}$$

Example:

- If Subodh is a BE student, then he loves programming.

- If Subodh loves programming, then he is expert in java.

∴ If Subodh is a BE student, then he is expert in java.

d. Disjunctive Syllogism Rule

$p \vee q$
$\neg p$
$\therefore q$

Example:

- Today is Wednesday or Thursday.
- Today is not Wednesday.

∴ Today is Thursday.

e. Addition rule

p
$\therefore p \vee q$

Example:

- Ram is a student of BE.

∴Ram is a student of BE or BCA.

f. Simplication rule

$p \wedge q$
$\therefore p$

Example:

-Subodh and Shyam are the students of BE.

∴Subodh is the student of BE

or

Shyam is the student of BE.

g. Conjunction rule

p
q
$\therefore p \wedge q$

Example:

- Shyam is the student of BE.
- Hari is the student of BE.

∴Shyam and Hari are the students of BE.

h. Resolution Rule

$p \vee q$

$\neg q \vee r$

$\therefore\ p \vee r$

Q. "If you send me an e-mail message then I will finish writing the program", "If you do not send me an e-mail message then I will go to sleep early", and "If I go to sleep early then I will wake up feeling refreshed". Lead to the conclusion "If I do not finish writing the program then I will wake up feeling refreshed".

Solution:

Let

p = "You send me an e-mail message"
q = "I will finish writing the program"
r = "I will go to sleep early"
s = "I will wake up feeling refreshed"

Hypothesis:

a. $p \rightarrow q$
b. $\neg p \rightarrow r$
c. $r \rightarrow s$

Conclusion: $\neg q \rightarrow s$

Steps	**Operations**	**Reasons**
1	$p \rightarrow q$	Given hypothesis
2	$\neg q \rightarrow \neg p$	Using contra positive on 1
3	$\neg p \rightarrow r$	Given hypothesis
4	$\neg q \rightarrow r$	Using hypothetical syllogism on 2 and 3
5	$r \rightarrow s$	Given hypothesis
6	$\neg q \rightarrow s$	Using hypothetical syllogism on 4 and 5

Hence the given hypotheses lead to the conclusion $\neg q \rightarrow s$

Q." Hari is playing in garden", "If he is playing in garden then he is not doing homework", "If he is not doing homework, then he is not learning" leads to the conclusion "He is not learning".

Solution:

Let

p = "Hari is playing in garden"
q = "He is doing homework"
r = "He is learning"

Hypothesis:

a. p
b. $p \rightarrow \neg q$
c. $\neg q \rightarrow \neg r$

Conclusion: $\neg r$

Steps	**Operations**	**Reasons**
1	p	Given hypothesis
2	$p \rightarrow \neg q$	Given hypothesis
3	$\neg q$	Using modus ponens on 1 and 2
4	$\neg q \rightarrow \neg r$	Given hypothesis
5	$\neg r$	Using modus ponens on 3 and 4

B. First Order Predicate Logic (FOPL)

Predicate (open proposition) is a mathematical logic that quantifies the variables in its formula with their common properties. This knowledge representation technique has three generic terms:

a. **Predicate or Propositional function**

   Predicate:

   Predicate is a part of declarative sentences describing the properties of an object or relation among objects. For example: “is a student” is a predicate as ‘A is a student’ and ‘B is a student’.

   Propositional function:

   Let p(x) be a statement involving a variable x and D is any set. We say that p is a predicate with respect to set D if for each x in D, p(x) is a proposition.

b. **Terms**

   Terms are any arguments in a predicate. The terms may be a constant, variable or any function.

   For example: “Hari’s father is Shyam’s father” = FATHER (FATHER (Hari), Shyam).

c. **Quantifier**

   Quantifiers are the tools to make the propositional function of a proposition. Construction of propositional function from predicators using quantifiers is called quantification.

   A quantifier is a symbol that permits one to declare the range or scope of variables in a logical expression. Two common quantifier are the existential quantifier. (“there exists or for some or at least one”) and universal quantifier. (“for all or for each or for any or for every and or for arbitrary”).

   Types of Quantifiers:

   a) Universal Quantifier (∀: For All)

      It is denoted by ∀ and used for universal quantification. The universal quantification of p(x) denoted by ∀x p(x) is proposition that is true for all values in universal set.

      The universal quantifier is read as:

      - For all x, p(x) holds
      - For each x, p(x) holds
      - For every x, p(x) holds

   b) Existential Quantifier(∃ : For Some)

It is denoted by ∃ and used for existential quantification. The existential quantification of p(x) denoted by $\exists x\ p(x)$ is proposition that is true for some values in universal set. The existential quantifier is read as:

- There is an x, such that p(x)
- There is at least one x such that p(x)
- For some x, p(x)

Propositional logic	**Predicate logic**
also called sentential logic	also called FOPL
includes sentence letters (A,B,C) and logical connectives	includes quantifies

Q. Assume that

P (x) denotes "x is an accountant."

Q (x) denotes "x owns a maruti."

Now, represent the following statement symbols.

a) All accountants own maruti.

Meaning: For all x, if x is an accountant, then x owns maruti

$\forall x$ p(x) q(x)

$\Rightarrow \forall x\ (\ p(x) \rightarrow\ q(x)\ )$

b) Some accountants own maruti

Meaning: For some x, x is an accountant and x owns maruti

$\exists x$ p(x) q(x)

$\Rightarrow \exists x\ (\ p(x) \wedge\ q(x)\ )$

c) All owners of maruti are accountants

Meaning: For all x, if x is a owner of maruti, then x is an accountant.

$\Rightarrow \forall x\ (\ q(x) \rightarrow\ p(x)\ )$

d) Someone who owns a maruti, is an accountant

Meaning: For some x, who owns a maruti and x is an accountant

$\Rightarrow \exists x\ (\ q(x) \wedge\ p(x)\ )$

Q. Convert into FOPL

a. All men are people.

$\Rightarrow \forall x$ MAN (x) $\rightarrow$ PEOPLE (x)

b. **Marcus was Pompeian.**

$\Rightarrow$POMPEIAN (Marcus)

c. **All Pompeian were Roman.**

$\Rightarrow \forall x$ POMPEIAN (x) $\rightarrow$ ROMAN (x)

d. **Ram tries to assassinate Hari.**

$\Rightarrow$ASSASSINATE (Ram, Hari)

e. **All Romans were either loyal to caser or hated him.**

$\Rightarrow \forall x$ ROMAN (x) $\rightarrow$ LOYAL (x, caser) $\vee$ HATES (x, caser)

f. **Socrates is a man. All men are mortal; therefore Socrates is mortal.**

$\Rightarrow$ MAN (Socrates), $\forall$ x MAN (x) $\rightarrow$ MORTAL (x), MORTAL (Socrates)

g. **Some student in this class has studied mathematics.**

$\Rightarrow$ Let

- S(x) = "x is a student in this class"
- M(x) = "x has studied mathematics"

Hence, required expression is: $\exists x\ [S(x) \wedge M(x)]$

Q. Convert into Well-Formed-Formula (WFF). Translate in two ways each of the following using predicates, quantifiers, and logical connectives. First, let the domain consists of the student in your class and second, let it consists of all people.

a. **Everyone in your class is friendly.**

$\Rightarrow$Let

- F(x) = "x is friendly"
- S(x) = "x is student in the class"

Domain	Well-Formed-Formulas (WFFs)
Student in the class	$\forall x\ F(x)$
All people	$\forall x\ [S(x) \rightarrow F(x)]$

b. **There is a person in your class who was not born in California**

$\Rightarrow$Let

- B(x) = "x born in California"
- S(x) = "x is student in the class"

Domain	Well-Formed-Formulas (WFFs)
Student in the class	$\neg\forall x \; B(x)$
All people	$\exists x \, [S(x) \land \neg B(x)]$

Rules of Inference for Quantified Statements

a. Universal Instantiation

$\forall x \; p(x)$
$\therefore p(d)$
Where d is in the domain of discourse D.

For example: If all balls in a box are red then any randomly drawn ball is also red .

b. Universal Generalization

$p(d)$
$\therefore \forall x \; p(x)$
Where, d is the domain of discourse D.

For example: If all the ball in a box are taken one by one in randomly manner and if all are red then we conclude that all balls in the box are red.

c. Existential Instantiations

$\exists x \; p(x)$
$\therefore p(d)$
For some d in the domain of discourse.

For example: If some balls in a box are red then resulting ball after experiment will also be red.

d. Existential Generalization

$p(d)$
$\therefore \exists x \; p(x)$
For some d in the domain of discourse.

For example: If we take only one random experiment for the ball drawn and apply the result of ball to the domain.

Q. Given Expression: All men are mortal. Einstein is a man. Prove that "Einstein is mortal" using FOPL.

Solution: Let

M(x) = "x is a man"

N(x) = "x is mortal"

Hypothesis: $\forall x \, [M(x) \rightarrow N(x)]$, M (Einstein)

Conclusion: N (Einstein)

Steps	Operations	Reasons
1.	$\forall x \, [M(x) \rightarrow N(x)]$	Given Hypothesis
2.	M(Einstein)→N (Einstein)	Using universal instantiation on 1
3.	M (Einstein)	Given Hypothesis

4.	N (Einstein)	Using modus pollens on 2 and 3

Hence the given hypotheses lead to the conclusion “Einstein is mortal”.

Q. Given Expression: “Lions are dangerous animals”, and “There are lions”. Prove that “There are dangerous animals” using FOPL.

Solution: Let

D(x) = “x is a dangerous animal”

L(x) = “x is a lion”

Hypothesis: $\forall x\ [L(x) \rightarrow D(x)]$, $\exists x\ L(x)$

Conclusion: $\exists x\ D(x)$

Steps	**Operations**	**Reasons**
1.	$\forall x\ [L(x) \rightarrow D(x)]$	Given Hypothesis
2.	$L(a) \rightarrow D(a)$	Using universal instantiation on 1
3.	$\exists x\ L(x)$	Given Hypothesis
4.	L(a)	Using existential instantiation on 3
5.	D(a)	Using modus pollenson 2 and 4
6.	$\exists x\ D(x)$	Using existential generalization on 5

Hence the given hypotheses lead to the conclusion “There are dangerous animals”

Q. Given Expression: “A student in this class has not read the book”, and “Everyone in this class passed the first exam”. Imply the conclusion “Someone has passed the exam has not read the book”.

Solution:

Let

C(x) = “x is in this class”

R(x) = “x has read the book”

P(x) = “x has passed the first exam”

Hypothesis: $\exists x\ [C(x) \land \neg R(x)]$, $\forall x\ [C(x) \rightarrow P(x)]$,

Conclusion: $\exists x\ [P(x) \land \neg R(x)]$

Steps	**Operations**	**Reasons**
1.	$\exists x\ [C(x) \land \neg R(x)]$	Given hypothesis
2.	$C(a) \land \neg R(a)$	Using existential instantiation on 1
3.	$\forall x\ [C(x) \rightarrow P(x)]$	Given hypothesis
4.	$C(a) \rightarrow P(a)$	Using universal instantiation on 3
5.	C(a)	Simplification on 2
6.	P(a)	Using modus penance on 4 and 5
7.	$\neg R(a)$	Simplification on 2
8.	$P(a) \land \neg R(a)$	Conjunction on 6 and 7
9.	$\exists x\ [P(x) \land \neg R(x)]$	Using existential generalization on 8

Hence the given hypotheses lead to the conclusion C(John).

Q. Differentiate between inference and reasoning.

Inference is a general term representing the derivation of new knowledge from existing knowledge and axioms (i.e., rules of derivation) within a single step, and can be one of many kinds, such as, induction, deduction and abduction. For example, "modus tollens" is a rule of inference. Thus, one inference is the derivation of new knowledge using a single step using modus tollens.

Reasoning is in context of a goal (e.g., decide whether a propositional formula is satisfiable or not) and is carried out via a search process involving multiple inferences. Choices during such search have to be made such as which axiom to "fire" along with which knowledge in order to derive new knowledge.

Resolution is a particular kind of reasoning involving the "resolution rule".

CNF

A sentence that is expressed as a conjunction of disjunctions of literals is said to be in conjunctive normal form (CNF). A sentence in CNF that contains only k literals per clause is said to be in k-CNF.

Conversion Procedure for CNF

We illustrate the procedure by converting the sentence $B \leftrightarrow (P \lor Q)$ into CNF. The steps are as follows:

Step 1: Eliminate $\leftrightarrow$, replacing $\alpha \leftrightarrow \beta$ with $(\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha)$.

$(B \rightarrow (P \lor Q)) \land ((P \lor Q) \rightarrow B)$

Step 2: Eliminate $\rightarrow$, replacing $\alpha \rightarrow \beta$ with $\neg\alpha \lor \beta$:

$(\neg B \lor P \lor Q) \land (\neg (P \lor Q) \lor B)$

Step 3: CNF requires $\neg$ to appear only in literals, so we “move $\neg$ inwards” by repeated application of the following equivalences:

→ $\neg(\neg\alpha) \equiv \alpha$ (double-negation elimination)

→ $\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$ (De Morgan)

→ $\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$ (De Morgan)

In the example, we require just one application of the last rule:

$(\neg B \lor P \lor Q) \land ((\neg P \land \neg Q) \lor B)$

Step 4: Now we have a sentence containing nested $\land$ and $\lor$ operators applied to literals. We apply the distributive law, distributing $\lor$ over $\land$ wherever possible.

i.e. $(\alpha \lor (\beta \land \gamma)) \equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))$ distributive of $\lor$ over $\land$

$(\neg B \lor P \lor Q) \land (\neg P \lor B) \land (\neg Q \lor B)$

The original sentence is now in CNF, as a conjunction of three clauses.

Resolution in Propositional Logic

Resolution principle was introduced by John Alan Robinson in 1965. The resolution technique can be applied for sentences in propositional logic and first-order logic. Resolution technique can be used only for disjunctions of literals to derive new conclusion. The resolution rule for the propositional calculus can be stated as following:

$(P \lor Q)$ and $(\neg Q \lor R)$, gives $(P \lor R)$.

Resolution refutation will terminate with the empty clause if it is logically equivalent (i.e. KB |= p). There are basic two methods for theorem proving using resolution, which are:

a. Forward chaining

- Forward chaining is one of the two main methods of reasoning when using inference rules
- Described logically as repeated application of modus ponens.
- Forward chaining is a popular implementation strategy for expert systems, business and production rule systems.
- Forward chaining starts with the available data and uses inference rules to extract more data until a goal is reached.
- An inference engine using forward chaining searches the inference rules until it finds one where the antecedent (If clause) is known to be true. When such a rule is found, the engine can conclude, or infer, the consequent (Then clause), resulting in the addition of new information to its data.

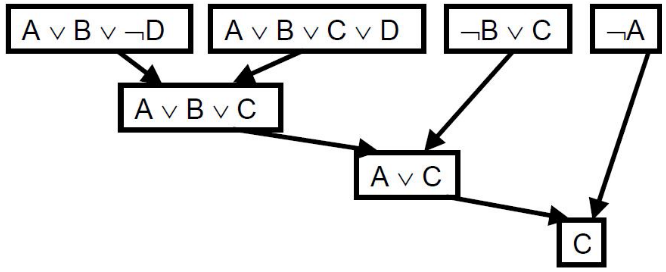
b. Backward chaining

- Backward chaining (or backward reasoning) is an inference method that can be described as working backward from the goal(s).
- In game theory, its application to sub games in order to find a solution to the game is called backward induction.
- In chess, it is called retrograde analysis, and it is used to generate table bases for chess end games for computer chess.
- Backward chaining is implemented in logic programming by SLD resolution
- Rules are based on the modus ponens inference rule.

Q. Let $P_1 = A \lor B \lor \neg D$, $P_2 = A \lor B \lor C \lor D$, $P_3 = \neg B \lor C$, $P_4 = \neg A$, $P_5 = C$ **then**

Show that $\{P_1, P_2, P_3, P_4\} \models P_5$

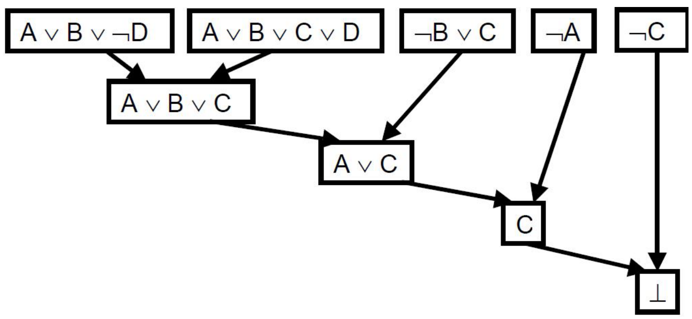
Solution:



Q. Let P_1 = A ∨B ∨¬D, P_2 = A ∨B ∨C ∨D, P_3 = ¬B ∨C, P_4 = ¬A, P_5= C then

Show that {P_1, P_2, P_3, P_4, ¬ P_5} |= ⊥

Solution:



Horn Clause

Horn clause is a disjunction of literals of which at most one (or only one) is positive. So all definite clauses are Horn clauses, as are clauses with no positive literals; these are called goal clauses. Horn clauses are closed under resolution: if we resolve two Horn clauses, we get back a Horn clause.

Resolution in FOPL

- **Unification Algorithm**

  During resolution in propositional logic, it is easy to determine that two literals (e.g. p and¬p) cannot both be true at the same time. In predicate logic this matching process is more complicated since the argument of the predicate must be considered.

  For example, MAN (John) and ¬MAN (John) is a contradiction, while MAN (John) and ¬MAN (Smith) is not. Thus, in order to determine contradictions, we need a matching procedure, called unification algorithm that compares two literals and discovers whether there exists a set of substitutions that makes them identical.

  To unify two literals, the initial predicate symbol on both must be same; otherwise there is no way of unification. For example, Q(x, y) and R(x, y) cannot unify but P(x, x) and P(y, z) can be unify by substituting z by x and y by x.

Q. Given Expression: John likes all kinds of foods. Apples are food. Chicken is food. Prove that John likes Peanuts using resolution.

Soln:

- FOPL

  → . ∀x FOOD (x) → LIKES (John, x) **or**

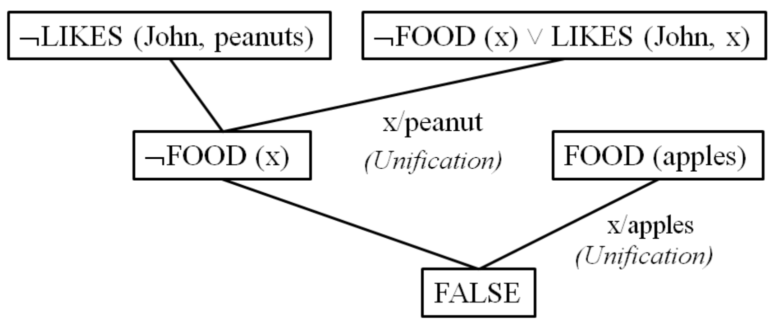
  ¬FOOD (x) ∨LIKES (John, x)

  → FOOD (apples)

  → FOOD (chicken)

Now, we have to prove: LIKES (John, peanuts). To prove the statement using resolution (proof by contradiction); let's take the negation of this as: ¬LIKES (John, peanuts)

Now,



Since, ¬LIKES (John, peanuts) is not possible and hence the: LIKES (John, peanuts) is proved.

Q. Given Expression: Bhaskar is a physician. All physicians know surgery. Prove that Bhaskar knows surgery using principle of resolution.

Soln:

- FOPL is

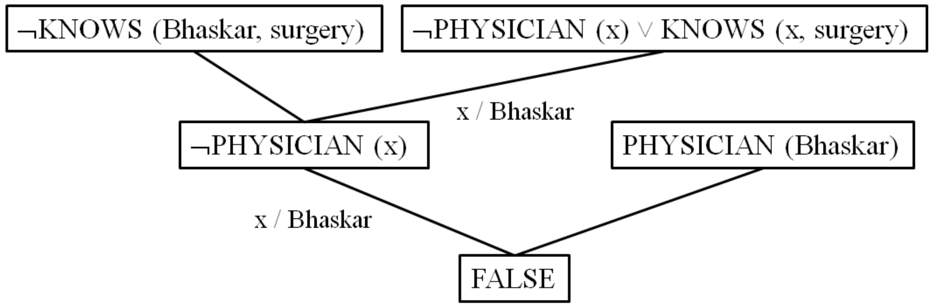
  → PHYSICIAN (Bhaskar)

  → . ∀x PHYSICIAN (x) → KNOWS (x, surgery) **or**

  ¬PHYSICIAN (x) ∨KNOWS (x, surgery)

Now, we have to prove that: KNOWS (Bhaskar, surgery). To prove the statement using resolution (proof by contradiction); let's take thenegation of this as: ¬KNOWS (Bhaskar, surgery)

Now,



Since, ¬KNOWS (Bhaskar, surgery) is not possible and hence the:

KNOWS(Bhaskar, surgery) is proved.

Q. Given Expression: All carnivorous animals have sharp teeth. Tiger is carnivorous. Fox is carnivorous. Prove that tiger has sharp teeth.

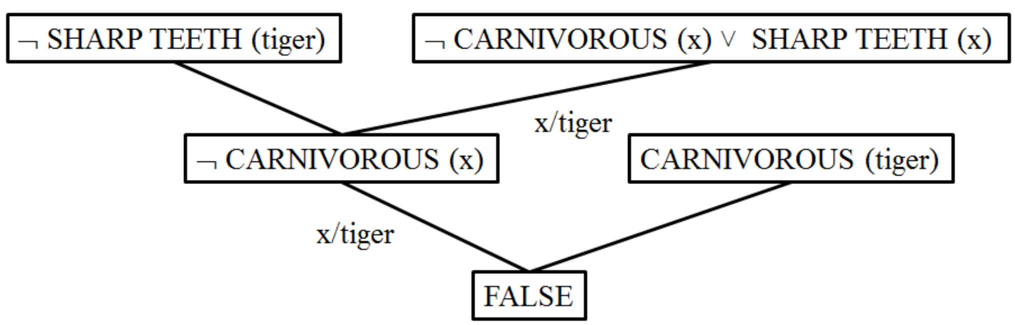
Soln:

- *FOPL is*
  - → ∀x CARNIVOROUS (x) → SHARP TEETH (x) **or**

    ¬CARNIVOROUS (x) ∨SHARP TEETH (x)
  - → CARNIVOROUS (tiger)
  - → CARNIVOROUS (fox)

Now, we have to prove that: SHARP TEETH (tiger). To prove the statement using resolution (proof by contradiction); let's take the negation of this as: ¬SHARPTEETH (tiger)

Now



Since, ¬SHARP TEETH (tiger) is not possible and hence the: SHARPTEETH (tiger) is proved.

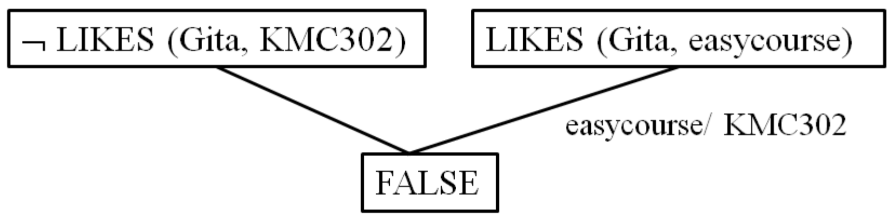
Q. Given Expression: Gita only likes easy course. Science courses are hard. All the courses in KMC are easy. KMC302 is a KMC course. Use the resolution to answer the question "Which course would Gita like?"

Soln:

- FOPL is
  → LIKES (Gita, easy course)
  → HARD COURSE (science)
  → ∀x KMC (x) → EASY COURSE (x) **or**
  ¬KMC (x) ∨EASY COURSE (x)
  → KMC (KMC302)

Now, we have to prove that: LIKES (Gita, KMC302). To prove the statementusing resolution (proof by contradiction); let's take the negation of this as: ¬LIKES (Gita, KMC302)

Now



Since, ¬LIKES (Gita, KMC302) is not possible and hence the: LIKES (Gita, KMC302) is proved.

Rule Based Deduction System

Rule-based systems are used as a way to store and manipulate knowledge to interpret information in a useful way. In this approach, idea is to use production rules, sometimes called IF-THEN rules. The syntax structure is

IF <premise> THEN <action>

- **<premise>**- is Boolean. The AND, and to a lesser degree OR and NOT, logical connectives are possible.
- **<action>**- a series of statements

A typical rule-based system has four basic components:

a. A **list of rules** or **rule base**, which is a specific type of knowledge base.
b. An **inference engine**, which infers information or takes action based on the interaction of input and the rule base.
c. Temporary **working memory**.
d. A **user interface** or other connection to the outside world through which input and output signals are received and sent.

Example: "If the patient has stiff neck, high fever and a headache, check for Brain Meningitis". Then it can be represented in rule based approach as:

IF *<fever, over, 39> and <neck, stiff, yes> and <head, pain, yes>*THEN

Add(<PATIENT,DIAGNOSE, MENINGITIS>)

Bayes Rule

Bayes rule can be useful for answering the probabilistic queries conditioned on one price of evidence. To compute just one conditional probability, it requires two terms:

a) A conditional probability and

b) Two unconditional probability.

Let A and B are two dependent events then the probability of the event A when the event B has already happened is called the conditional probability. It is denoted by P (A|B) and is given by:

$P(A|B) = P(A \cap B)/P(B)$, where $P(B) \neq 0$, $\Rightarrow P(A \cap B) = P(A|B).\ P(B)$ ---- (i)

Similarly,

$P(B|A) = P(A \cap B)/P(A)$, where $P(A) \neq 0$, $\Rightarrow P(A \cap B) = P(B|A).\ P(A)$ ---- (ii)

From equation (i) and (ii), we have

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

Bayes rule is useful for those cases where P (A|B) can be estimated but P (B|A) is hard to find experimentally.

In a task such as medical diagnosis, we often have conditional probabilities on causal relationships and want to derive a diagnosis. A doctor knows the probability of symptoms condition to disease P (S|D), and the patient knows his own feeling or symptoms P(S). Also the doctor knows about probability of disease P (D), then the probability of disease condition to symptoms can be defined as:

For example:

Let,

i) S = Symptoms on patients such as stiff neck whose probability P(S) is =1/20

ii) D = Disease known by doctor whose probability P (D) is = 1/50000

iii) the given P (S|D) = 0.5 or 50%

Now, the probability of disease condition to symptoms,

$P(D|S)$

$= [P(S|D).\ P(D)]/P(S)$

$= 0.0002$

Assignment

Why probabilistic reasoning is important in AI? Explain with Example.

Causal Networks

A causal network is an acyclic (not cyclic) directed graph arising from an evolution of a substitution system. The substitution system is a map which uses a set of rules to transform elements of a sequence into anew sequence using a set of rules which "translate" from the original sequence to its transformation. For example, the substitution system 1→0, 0→11 would take 10→011 →1100→ 001111 → 11110000 → …. ……..

A causal network is a Bayesian network with an explicit requirement that the relationships be causal.

Reasoning in Belief Networks

A Bayesian network, Bayes network, belief network or probabilistic directed acyclic graphical model is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG).

For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. From given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

Formally, Bayesian networks are directed acyclic graphs whose nodes represent random variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes which are not connected represent variables which are conditionally independent of each other. Each node is associated with a probability function that takes as input a particular set of values for the node's parent variables and gives the probability of the variable represented by the node. Bayesian networks are used for modeling knowledge in computational biology, medicine, document classification, information retrieval, semantic search, image processing, data fusion, decision support systems, engineering, and gaming.