Unit 4 8 hrs.

Knowledge Representation, Inference and Reasoning

4.1 Approaches to Knowledge Representation
4.2 Issues in Knowledge Representation
4.3 Propositional Logic, Predicate Logic, FOPL
4.4 Rules of Inference, Resolution Refutation System (RRS), Answer Extraction from RRS
4.5 Statistical Reasoning:
Probability and Bayes Theorem and Causal Networks, Reasoning in Belief Network
4.6 Semantic Nets and Frames

Knowledge Representation:

Humans are best at understanding, reasoning, and interpreting knowledge. Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world. But how machines do all these things comes under knowledge representation and reasoning. Hence, we can describe Knowledge representation as following:

- Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behavior of agents.
- It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real-world problems such as diagnosis a medical condition or communicating with humans in natural language.
- It is also a way which describes how we can represent knowledge in artificial intelligence. Knowledge representation 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.

Basic Terminology

Object: All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments.

Events: Events are the actions which occur in our world.

Performance: It describe behavior which involves knowledge about how to do things.

Department of Computer Science – DEC [NAST]

Meta-knowledge: It is knowledge about what we know.

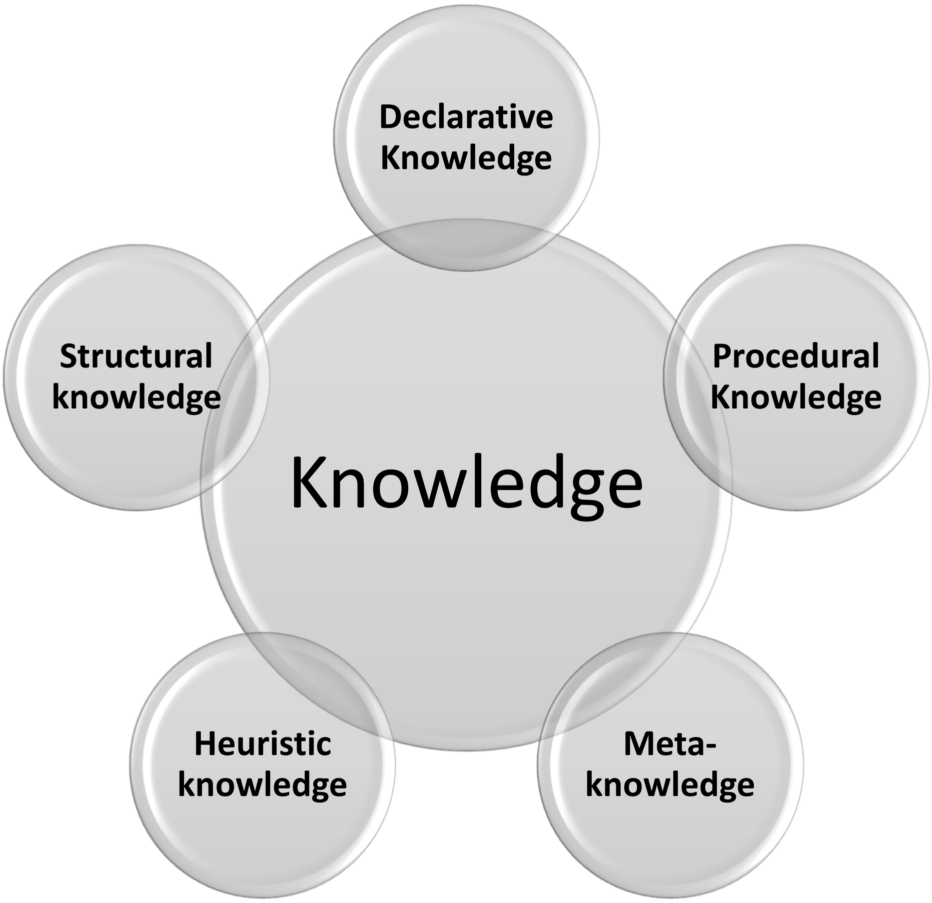
Facts: Facts are the truths about the real world and what we represent.

Knowledge-Base: The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).

Knowledge: Knowledge is awareness or familiarity gained by experiences of facts, data, and situations. Following are the types of knowledge in artificial intelligence:

Types of knowledge

Following are the various types of knowledge:



1. Declarative Knowledge:

- Declarative knowledge is to know about something.
- It includes concepts, facts, and objects.
- It is also called descriptive knowledge and expressed in declarative sentences.
- It is simpler than procedural language.

2. Procedural Knowledge

- It is also known as imperative knowledge.
- Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.
- It can be directly applied to any task.
- It includes rules, strategies, procedures, agendas, etc.
- Procedural knowledge depends on the task on which it can be applied.

3. Meta-knowledge:

- Knowledge about the other types of knowledge is called Meta-knowledge.

4. Heuristic knowledge:

- Heuristic knowledge is representing knowledge of some experts in a filed or subject.
- Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.

5. Structural knowledge:

- Structural knowledge is basic knowledge to problem-solving.
- It describes relationships between various concepts such as kind of, part of, and grouping of something.
- It describes the relationship that exists between concepts or objects.

Department of Computer Science – DEC [NAST]

4.1 Approaches to Knowledge Representation

There are mainly four approaches to knowledge representation, which are given below:

1. Simple relational knowledge:

- It is the simplest way of storing facts which uses the relational method, and each fact about a set of the object is set out systematically in columns.
- This approach of knowledge representation is famous in database systems where the relationship between different entities is represented.
- This approach has little opportunity for inference.

Example: The following is the simple relational knowledge representation.

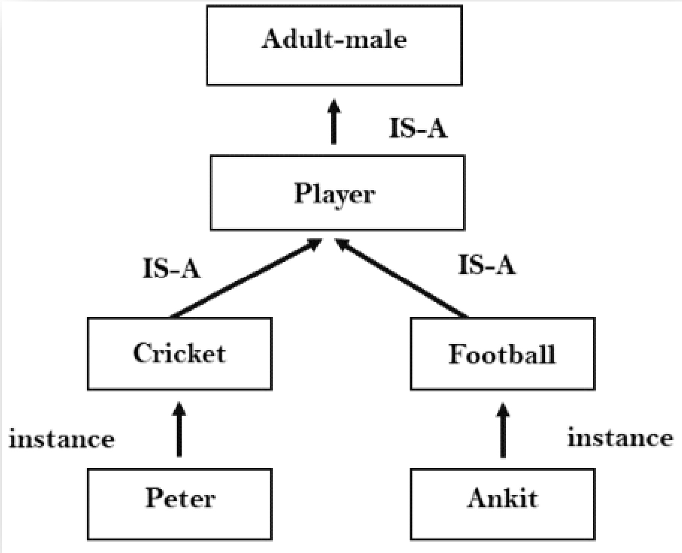
Player	Weight	Age
Player1	65	23
Player2	58	18
Player3	75	24

2. Inheritable knowledge:

- In the inheritable knowledge approach, all data must be stored into a hierarchy of classes.
- All classes should be arranged in a generalized form or a hierarchal manner.
- In this approach, we apply inheritance property.
- Elements inherit values from other members of a class.
- This approach contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation.
- Every individual frame can represent the collection of attributes and its value.
- In this approach, objects and values are represented in Boxed nodes.
- We use Arrows which point from objects to their values.

Department of Computer Science – DEC [NAST]

- **Example:**



3. Inferential knowledge:

- Inferential knowledge approach represents knowledge in the form of formal logics.
- This approach can be used to derive more facts.
- It guaranteed correctness.
- **Example:** Let's suppose there are two statements:
  a. **Marcus is a man**
  b. **All men are mortal**

  Then it can represent as;

a.	**man (Marcus)**
b.	**∀x = man (x) ----------> mortal (x)**

4. Procedural knowledge:

- Procedural knowledge approach uses small programs and codes which describes how to do specific things, and how to proceed.
- In this approach, one important rule is used which is **If-Then rule**.
- In this knowledge, we can use various coding languages such as **LISP language** and **PROLOG language**.
- We can easily represent heuristic or domain-specific knowledge using this approach.
- But it is not necessary that we can represent all cases in this approach.

Requirements for knowledge Representation system:

A good knowledge representation system must possess the following properties.

1. **Representational Accuracy:**
   KR system should have the ability to represent all kind of required knowledge.

2. **Inferential Adequacy:**
   KR system should have ability to manipulate the representational structures to produce new knowledge corresponding to existing structure.

3. **Inferential Efficiency:**
   The ability to direct the inferential knowledge mechanism into the most productive directions by storing appropriate guides.

4. **Acquisitional efficiency:**
   The ability to acquire the new knowledge easily using automatic methods.

4.2 Issues in Knowledge Representation

Issues that arises while Knowledge Representation techniques:

1. Important attributes.
2. Relationships among attributes.
3. Choosing the granularity of representation.
4. Representing sets of objects.
5. Finding the right structure as needed

Department of Computer Science – DEC [NAST]

4.3 Propositional Logic, Predicate Logic, FOPL

In order to solve complex problems in AI, we need

- Large amount of knowledge.
- Mechanism to manipulate knowledge to create solutions to new problem.

Methods for knowledge representation:

1. Logical Representation
    a. Propositional Logic
    b. Predicate Logic or FOPL (First Order Predicate Logic)
2. Production Rule
3. Semantic Networks
4. Frame Representation

Propositional Logic

➔ Propositional is a statement of a fact.

➔ Propositional is a declarative sentence whose value is either **true** (denoted either T or 1) or **false** (denoted either F or 0).

➔ Variables are used to represent propositions.

➔ Propositional Symbols/Variables a, b, and c, or

➔ The most common variables used are p, q, and r.

➔ Examples:

- *"Today is Sunday".* **[Atomic Proposition]**
- *"Today is Sunday and it's sunny day.* **[Molecular/Complex Proposition]**
- *"10+10=20"* **[Atomic Proposition]**

➔ Ram is an honest boy. ---- **Fact {Propositional** — T or F}

➔ Is Ram an honest boy? ---- {It fails to be a propositional}

Department of Computer Science – DEC [NAST]

Some Terminologies in Propositional Algebra

- **Statement:** Sentence that can be true / false
- **Properties of Statements:**
    - **Satisfiability:** a sentence is satisfiable if there is an interpretation for which it is true. A sentence is true if it is true in some model
        - **E.g. *"Today is cold climate"***
    - **Contradiction:** if there is no interpretation for which sentence is true.
        - **E.g. *"Japan is capital of India"***
    - **Validity:** a sentence is valid if it is true for every interpretation. Valid sentences are also known as **Tautology.**
        - **E.g. *"Kathmandu is the capital of Nepal"***

Logical Operators

- **Unary Operator**
    - Negation: *"NOT p"*, $\neg p, \sim p$
- **Binary Operators**
    - Conjunction: "p AND q", $p \wedge q$
    - Disjunction: "p OR q", $p \vee q$
    - Implication: "IF p THEN q", $p \rightarrow q$
    - Biconditional: "p IF AND ONLY IF q", $p \leftrightarrow q$

Unary Operator: Negation

- Negation Operator, "NOT", has symbol $\neg$ $\boldsymbol{or}$ $\sim$
- **Example:**
    - p: Today is sunny day.
    - **¬p** can be read as:
        - Today is not sunny day
        - Today is cold day.
        - Today is cloudy day.
        - Today is rainy day.
    - **Truth Table:**

p	$\neg p$
T	F
F	T

© Deepak Bhatta Kaji (MCA)
Department of Computer Science – DEC [NAST]

Binary Operator: Conjunction

- Conjunction Operator, "AND", has symbol ∧
- **Example:**
  - p: Today is sunny day.
  - q: I am eating an Ice-cream.
  - $\boldsymbol{p \wedge q}$ can be read as:
    - Today is sunny day, and I am eating Ice-cream.
  - **Truth Table:**

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Binary Operator: Disjunction

- Conjunction Operator, "OR", has symbol ∨
- **Example:**
  - p: Today is sunny day.
  - q: I am eating an Ice-cream.
  - $\boldsymbol{p \vee q}$ can be read as:
    - Today is sunny day, or I am eating Ice-cream.
  - **Truth Table:**

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Binary Operator: Implication

- Implication Operator, "IF…THEN…", has symbol →
- **Example:**
  - p: It is raining outside.
  - q: The road is wet.
  - $\boldsymbol{p \rightarrow q}$ can be read as:
    - If it is raining outside, then the road is wet.
- For the compound statement $\boldsymbol{p \rightarrow q}$
  - $\boldsymbol{p}$ is called premise, hypothesis or the antecedent.
  - $\boldsymbol{q}$ is called the conclusion or consequent.
  - $\boldsymbol{q \rightarrow p}$ is the converse of $p \rightarrow q$
  - $\boldsymbol{\neg q \rightarrow \neg p}$ is the contrapositive of $\boldsymbol{p \rightarrow q}$

Department of Computer Science – DEC [NAST]

- **Truth Table:**

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Binary Operator: Biconditional

- Biconditional Operator, "IF and only IF", has symbol $\boldsymbol{p \leftrightarrow q}$
- **Example:**
    - p: It is raining outside.
    - q: The road is wet.
    - $\boldsymbol{p \leftrightarrow q}$ can be read as:
        - If it is raining outside if and only if the road is wet.
    - The biconditional statement is equivalent to $\boldsymbol{(p \to q) \wedge (q \to p)}$.
    - In other word, for $\boldsymbol{p \leftrightarrow q}$ to be true we must have both p and q true or both false.

    - **Truth Table:**

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

Using truth table, show that $p \leftrightarrow q$ is logically equivalent to $(p \to q) \wedge (q \to p)$

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

Predicate Logic or FOPL

In the topic of Propositional logic, we have seen that how to represent statements using propositional logic. But unfortunately, in propositional logic, we can only represent the facts, which are either true or false. PL is not sufficient to represent the complex sentences or natural language statements. The propositional logic has very limited expressive power. Consider the following sentence, which we cannot represent using PL logic.

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

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

- First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.
- FOL is sufficiently expressive to represent the natural language statements in a concise way.
- First-order logic is also known as **Predicate logic or First-order predicate logic**. First-order logic is a powerful language that develops information about the objects in an easier way and can also express the relationship between those objects.
- First-order logic (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world:
    - **Objects:** A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, ...
    - **Relations: It can be unary relation such as:** red, round, is adjacent, **or n-any relation such as:** the sister of, brother of, has color, comes between
    - **Function:** Father of, best friend, third inning of, end of, ...
- As a natural language, first-order logic also has two main parts:
    a. **Syntax**
    b. **Semantics**

Syntax of First-Order logic:

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

Department of Computer Science – DEC [NAST]

Basic Elements of First-order logic:

Following are the basic elements of FOL syntax:

Constant	1, 2, a, john, mumbai, cat, dog ....
Variables	X, Y, Z, A, B, X1, X2 ....
Predicates	brother, father ….
Function	sqrt, leftLegOf, ...
Connectives	∧, ∨, ¬, ⇒, ⇔
Equality	==
Quantifier	∀ *(for All)*, ∃ *(there Exist)*

Atomic sentences:

- Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.
- We can represent atomic sentences as **Predicate (term1, term2, ..., term n)**.

Example: Ram and Lakshman are brothers: => brothers(ram, lakshman).
munna is a cat: => cat (munna).

Complex Sentences:

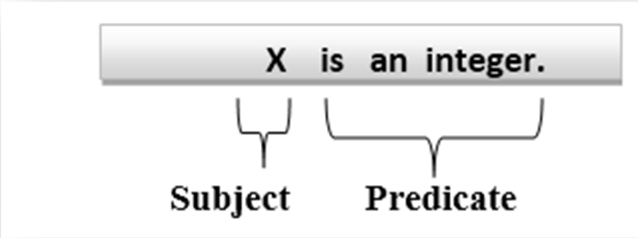
- Complex sentences are made by combining atomic sentences using connectives.

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

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

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

Department of Computer Science – DEC [NAST]



Quantifiers in First-order logic:

- A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.
- These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression. There are two types of quantifier:
    a. (∀) **Universal Quantifier, (for all, everyone, everything)**
    b. (∃) **Existential quantifier, (for some, at least one).**

Universal Quantifier:

Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.

The Universal quantifier is represented by a symbol ∀, which resembles an inverted A.

Note: In universal quantifier we use implication "→".

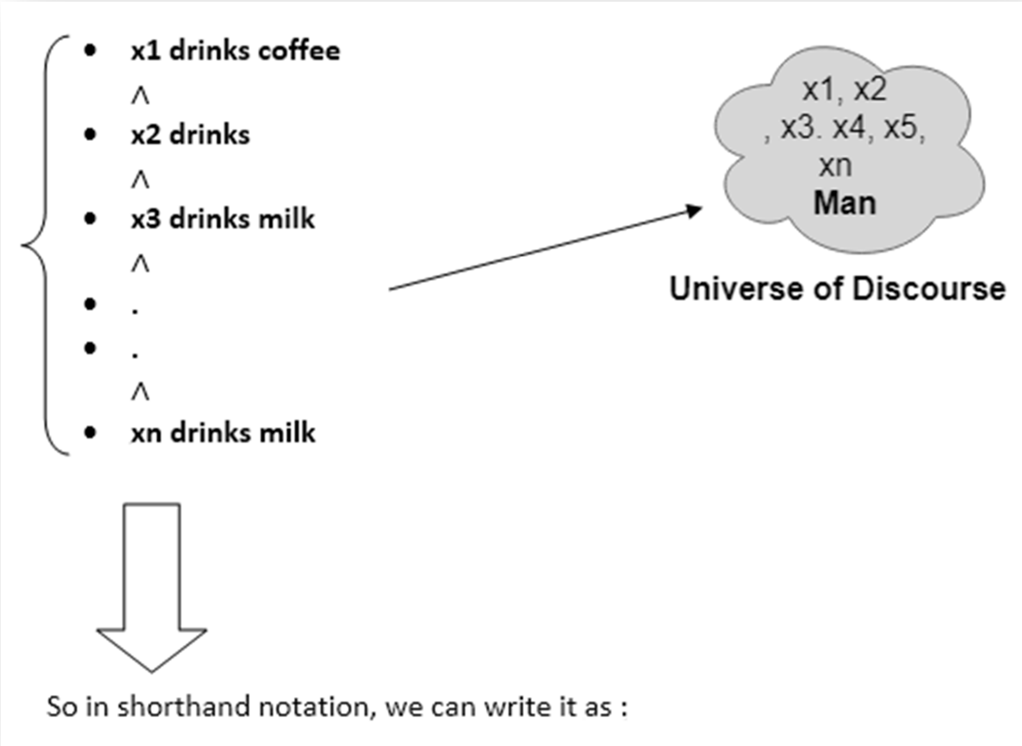
If x is a variable, then ∀x is read as:

- **For all x**
- **For each x**
- **For every x.**

Example:

All man drink coffee.

Let a variable x which refers to a cat so all x can be represented in UOD as below:



∀x man(x) → drink (x, coffee).

It will be read as: There are all x where x is a man who drink coffee.

Existential Quantifier:

Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.

It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.

> ***Note: In Existential quantifier we always use AND or Conjunction symbol (∧).***

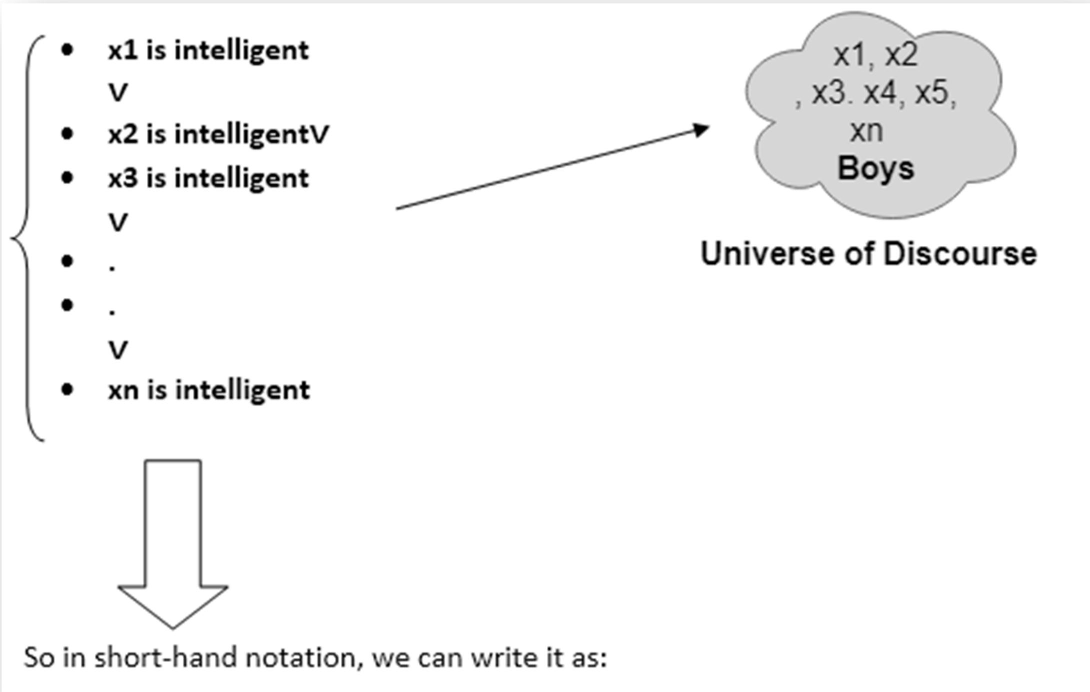
If x is a variable, then existential quantifier will be ∃x or ∃(x). And it will be read as:

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

Department of Computer Science – DEC [NAST]

Example:

Some boys are intelligent.



∃x: boys(x) ∧ intelligent(x)

It will be read as: There are some x where x is a boy who is intelligent.

Points to remember:

- The main connective for universal quantifier ∀ is implication →.
- The main connective for existential quantifier ∃ is and ∧.

Properties of Quantifiers:

- In universal quantifier, ∀x∀y is similar to ∀y∀x.
- In Existential quantifier, ∃x∃y is similar to ∃y∃x.
- ∃x∀y is not similar to ∀y∃x.

Department of Computer Science – DEC [NAST]

Some Examples of FOL/FOPL using quantifier:

1. All birds fly.

In this question the predicate is "**fly(bird)**."

And since there are all birds who fly so it will be represented as follows.

∀x: bird(x) →fly(x).

2. Every man respects his parent.

In this question, the predicate is "**respect(x, y)," where x=man, and y= parent**.

Since there is every man so will use ∀, and it will be represented as follows:

∀x: man(x) → respects (x, parent).

3. Some boys play cricket.

In this question, the predicate is "**play(x, y)**," where x= boys, and y= game. Since there are some boys so we will use ∃**, and it will be represented as**:

∃x: boys(x) → play(x, cricket).

4. Not all students like both Mathematics and Science.

In this question, the predicate is "**like(x, y)," where x= student, and y= subject**.

Since there are not all students, so we will use **∀ with negation, so** following representation for this:

¬∀ (x): [student(x) → like(x, Mathematics) ∧ like(x, Science)].

5. Only one student failed in Mathematics.

In this question, the predicate is "**failed(x, y)," where x= student, and y= subject**.

Since there is only one student who failed in Mathematics, so we will use following representation for this:

∃(x): [student(x) → failed (x, Mathematics) ∧∀ (y) [¬(x==y) ∧ student(y) → ¬failed (x, Mathematics)].

Rules of Inference

Idempotent rule: $p \wedge p \equiv p$ $p \vee p \equiv p$	**Commutative rule:** $p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$
Associative rule: $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$ $p \vee (p \vee r) \equiv (p \vee q) \vee r$	**Distributive rule:** $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (p \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
De-Morgan's rule: $\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	**Contrapositive rule:** $p \rightarrow q \equiv \neg q \rightarrow \neg p$
Implication Elimination: $p \rightarrow q \equiv \neg p \vee q$	**Bidirectional Implication Elimination:** $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
Double Negation rule: $\neg(\neg p) \equiv p$	**Absorption rule:** $p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$
Fundamental Identities: $p \wedge \neg p \equiv F\ [Contradiction]$ $p \vee \neg p \equiv T\ [Tautology]$ $p \vee T \equiv p$ $p \vee F \equiv p$ $p \wedge T \equiv p$ $p \wedge F \equiv F$	**Modus Ponens:** **-**If p is **true** and $p \rightarrow q$ then we infer q is also **true** p $p \rightarrow q$ Hence, q *E.g: "If Tweety is a bird then Tweety filies."* *"Tweety is a bird."* *Therefore, Tweety flies.* **Modus Ponens:** **-**If ¬p is **true** and $p \rightarrow q$ then we infer ¬q is also **true** $\neg p$ $p \rightarrow q$ Hence, ¬q *E.g: "If Tweety is a bird then Tweety filies."* *"Tweety doesn't fly."* *Therefore, Tweety is not a bird.*

Department of Computer Science – DEC [NAST]

Resolution Refutation System (RRS)

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., **proofs by contradictions**. It was invented by a Mathematician **John Alan Robinson** in the year 1965.

Resolution 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 proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the conjunctive normal form or clausal form.

Clause: Disjunction of literals (an atomic sentence) is called a clause. It is also known as a unit clause.

Conjunctive Normal Form: A sentence represented as a conjunction of clauses is said to be conjunctive normal form or CNF.

The resolution rule for first-order logic is simply a lifted version of the propositional rule. Resolution can resolve two clauses if they contain complementary literals, which are assumed to be standardized apart so that they share no variables.

Where $\mathbf{l_i}$ and $\mathbf{m_j}$ are complementary literals.

This rule is also called the **binary resolution rule** because it only resolves exactly two literals.

Example:

We can resolve two clauses which are given below:

[Animal (g(x) V Loves (f(x), x)]   and   [¬ Loves(a, b) V ¬Kills(a, b)]

Where two complimentary literals are: **Loves (f(x), x) and ¬ Loves (a, b)**

These literals can be unified with unifier θ= [a/f(x), and b/x], and it will generate a resolvent clause:

[Animal (g(x) V ¬ Kills(f(x), x)].

Steps for Resolution:

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

To better understand all the above steps, we will take an example in which we will apply resolution.

Example:

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.

Prove by resolution that: **John likes peanuts.**

Solution:

Step-1: Conversion of Facts into FOL

Step-2: Convert FOL statements into CNF

In the first step we will convert all the given statements into its first order logic and then convert it into CNF.

In First order logic resolution, it is required to convert the FOL into CNF as CNF form makes easier for resolution proofs.

- Eliminate all implication (→) and rewrite.
- Move negation (¬) inwards and rewrite.
- Rename variables or standardize variables.
- Eliminate existential instantiation quantifier by elimination.
- Drop Universal quantifiers.

a. John likes all kind of food.

FOL: $\forall \boldsymbol{X}: \boldsymbol{food}(\boldsymbol{X}) \rightarrow \boldsymbol{likes}(\boldsymbol{john}, \boldsymbol{X})$

CNF: $\neg \boldsymbol{food}(\boldsymbol{X}) \vee \boldsymbol{likes}(\boldsymbol{john}, \boldsymbol{X})$

b. Apple and vegetable are food

FOL: $\boldsymbol{food}(\boldsymbol{apple}) \wedge \boldsymbol{food}(\boldsymbol{vegetables})$

CNF: $\boldsymbol{i}) \ \boldsymbol{food}(\boldsymbol{apple}) \qquad \boldsymbol{ii}) \ \boldsymbol{food}(\boldsymbol{vegetables})$

c. Anything anyone eats and not killed is food.

FOL: $\forall \boldsymbol{X}, \forall \boldsymbol{Y}: \boldsymbol{eats}(\boldsymbol{X}, \boldsymbol{Y}) \wedge \neg \boldsymbol{killed}(\boldsymbol{X}) \rightarrow \boldsymbol{food}(\boldsymbol{Y})$

CNF: $\neg \boldsymbol{eats}(\boldsymbol{X1}, \boldsymbol{Y1}) \vee \boldsymbol{killed}(\boldsymbol{X1}) \vee \boldsymbol{food}(\boldsymbol{Y1})$

d. Anil eats peanuts and still alive

FOL: $\boldsymbol{eats}(\boldsymbol{anil}, \boldsymbol{peanuts}) \wedge \boldsymbol{alive}(\boldsymbol{anil})$

CNF: $\boldsymbol{eats}(\boldsymbol{anil}, \boldsymbol{peanuts}) \wedge \boldsymbol{alive}(\boldsymbol{anil})$ **or**

$\boldsymbol{eats}(\boldsymbol{anil}, \boldsymbol{peanuts}) \wedge \neg \boldsymbol{killed}(\boldsymbol{anil})$

$\boldsymbol{i}) \; \boldsymbol{eats}(\boldsymbol{anil}, \boldsymbol{peanuts}) \quad \boldsymbol{ii}) \; \neg \boldsymbol{killed}(\boldsymbol{anil})$

e. Harry eats everything that Anil eats.

FOL: $\forall \boldsymbol{X}: \boldsymbol{eats}(\boldsymbol{anil}, \boldsymbol{X}) \rightarrow \boldsymbol{eats}(\boldsymbol{harry}, \boldsymbol{X})$

CNF: $\neg \boldsymbol{eats}(\boldsymbol{anil}, \boldsymbol{X2}) \vee \boldsymbol{eats}(\boldsymbol{harry}, \boldsymbol{X2})$

f. John likes Peanuts *(proven statement)*

FOL: $\boldsymbol{likes}(\boldsymbol{john}, \boldsymbol{peanuts})$

CNF: $\boldsymbol{likes}(\boldsymbol{john}, \boldsymbol{peanuts})$

Note: Statements "food(Apple) Λ food(vegetables)" and "eats (Anil, Peanuts) Λ alive(Anil)" can be written in two separate statements.

Step-3: Negate the statement to be proved

In this statement, we will apply negation to the conclusion statements, which will be written as: **¬likes(john, peanuts)**

Step-4: Draw Resolution graph:

Now in this step, we will solve the problem by resolution tree using substitution. For the above problem, it will be given as follows:

Department of Computer Science – DEC [NAST]

$\neg \cancel{likes}(\cancel{john}, peanuts)$ $\qquad$ $\neg food(X) \lor \cancel{likes}(\cancel{john}, X)$

{peanuts/X}

$\neg \cancel{food}(peanuts)$ $\qquad$ $\neg eats(X1, Y1) \lor killed(X1) \lor \cancel{food}(Y1)$

{peanuts/Y1}

$\neg \cancel{eats}(X1, \cancel{peanuts}) \lor killed(X1)$ $\qquad$ $\cancel{eats}(anil, \cancel{peanuts})$

{anil/X1}

$\cancel{killed}(\cancel{anil})$ $\qquad$ $\neg \cancel{killed}(\cancel{anil})$

{ EMPTY } Hence, Proved

Conclusion: - $\neg likes(john, peanuts)$ is a false.

Therefore, $likes(john, peanuts)$ must be True.

Hence, John likes Peanuts. Proved.

Answer Extraction from RRS

Problem: How to extract answers from the given facts.

Assume the following facts:

i. Kabita like easy courses.
ii. Science Courses are hard.
iii. All the course in the Computer Engineering Department are easy.
iv. Com101 is a Computer Engineering Course.

Use resolution to answer the question "What course would Kabita like?"

Sol^n:

i. Kabita like easy courses.

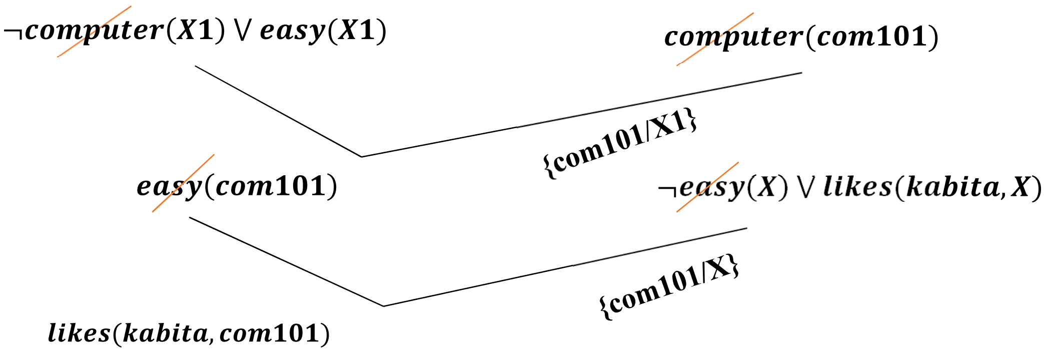
FOL: $\forall X: \boldsymbol{easy(X) \rightarrow likes(kabita, X)}$
CNF: $\neg \boldsymbol{easy(X)} \vee \boldsymbol{likes(kabita, X)}$

ii. Science Courses are hard.
FOL: $\boldsymbol{hard(science)} \quad \equiv \quad \neg \boldsymbol{easy(science)}$
CNF: $\neg \boldsymbol{easy(science)}$

iii. All the course in the Computer Engineering Department are easy.
FOL: $\forall X: \boldsymbol{computer(X) \rightarrow easy(X)}$
CNF: $\neg \boldsymbol{computer(X1)} \vee \boldsymbol{easy(X1)}$ *//use different variable names*

iv. Com101 is a Computer Engineering Course.
FOL: $\boldsymbol{computer(com101)}$
CNF: $\boldsymbol{computer(com101)}$

Draw Resolution graph:



Likes(kabita,com101)

Thus, Kabita likes Com101

Using the resolution we can extract the answer of question.

4.5 Statistical Reasoning:

Till now, we have learned knowledge representation using first-order logic and propositional logic with certainty, which means we were sure about the predicates. With this knowledge representation, we might write A→B, which means if A is true then B is true, but consider a situation where we are not sure about whether A is true or not then we cannot express this statement, this situation is called uncertainty.

So, to represent uncertain knowledge, where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning.

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.
We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of someone's laziness and ignorance.
In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today," "behavior of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

Need of probabilistic reasoning in AI:

- When there are unpredictable outcomes.
- When specifications or possibilities of predicates becomes too large to handle.
- When an unknown error occurs during an experiment.

In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

- **Bayes' rule**
- **Bayesian Statistics**

Probability and Bays Theorem

Probability: Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

- ✓ $0 \leq P(A) \leq 1$, where P(A) is the probability of an event A.
- ✓ P(A) = 0, indicates total uncertainty in an event A.
- ✓ P(A) =1, indicates total certainty in an event A.

We can find the probability of an uncertain event by using the below formula.

$$\textbf{Probability of occurrence} = \frac{\text{Number of desired outcomes}}{\text{Total number of outcomes}}$$

- ○ $P(\neg A)$ = probability of a not happening event.
- ○ $P(\neg A) + P(A) = 1$.

Event: Each possible outcome of a variable is called an event.

Sample space: The collection of all possible events is called sample space.

Random variables: Random variables are used to represent the events and objects in the real world.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior Probability: The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

Conditional probability:

Conditional probability is a probability of occurring an event when another event has already happened.

Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

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

Where P($A \wedge B$)= Joint probability of a and B

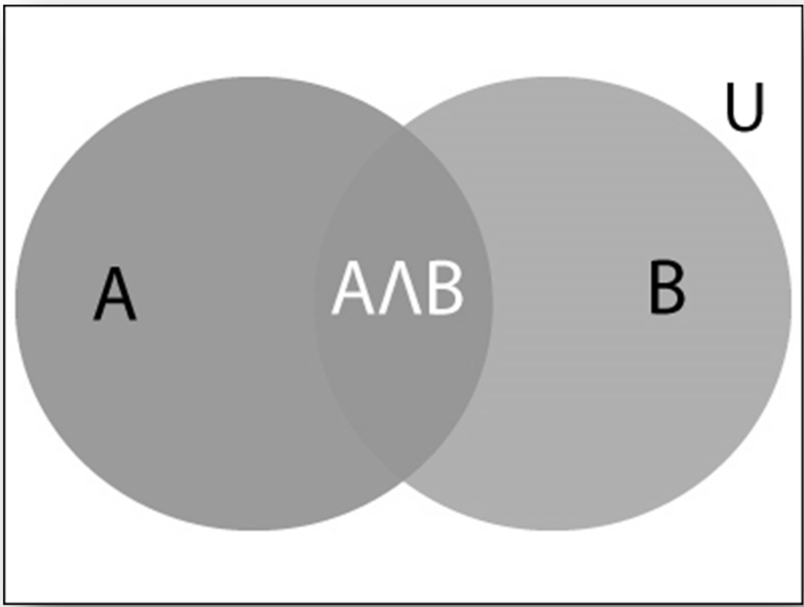
P(B)= Marginal probability of B.

Department of Computer Science – DEC [NAST]

If the probability of A is given and we need to find the probability of B, then it will be given as:

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

It can be explained by using the below Venn diagram, where B is occurred event, so sample space will be reduced to set B, and now we can only calculate event A when event B is already occurred by dividing the probability of **P(A⋀*B*) by P(B)**.



Example:

In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like English also like mathematics?

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{0.4}{0.7} = 57\%$$

Hence, 57% are the students who like English also like Mathematics.

Bayes' theorem:

Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.

Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.

It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).

Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Example: If cancer corresponds to one's age then by using Bayes' theorem, we can determine the probability of cancer more accurately with the help of age.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:

As from product rule we can write:

P(A ∧ B)= P(A|B) P(B) or

Similarly, the probability of event B with known event A:

P(A ∧ B)= P(B|A) P(A)

Equating right hand side of both the equations, we will get:

$$P(A|B) = \frac{P(B|A)\, P(A)}{P(B)} \quad \text{....(a)}$$

The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

It shows the simple relationship between joint and conditional probabilities. Here,

P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

P(B|A) is called the **likelihood**, in which we consider that hypothesis is true, then we calculate the probability of evidence.

P(A) is called the **prior probability**, probability of hypothesis before considering the evidence

P(B) is called **marginal probability**, pure probability of an evidence.

In the equation (a), in general, we can write P (B) = P(A)*P(B|Ai), hence the Bayes' rule can be written as:

$$P(A_i|B) = \frac{P(A_i)*P(B|A_i)}{\sum_{i=1}^{k} P(A_i)*P(B|A_i)}$$

Where $A_1, A_2, A_3, \ldots\ldots, A_n$ is a set of mutually exclusive and exhaustive events.

Applying Bayes' rule:

Bayes' rule allows us to compute the single term P(B|A) in terms of P(A|B), P(*B*), and P(A). This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

$$P(\text{cause}|\text{effect}) = \frac{P(\text{effect}|\text{cause})\, P(\text{cause})}{P(\text{effect})}$$

Application of Bayes' theorem in Artificial intelligence:

Following are some applications of Bayes' theorem:

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.
- It can solve the Monty Hall problem.

Department of Computer Science – DEC [NAST]

Example of Bayes' theorem

Q) An insurance company insured 2000 scooter driver, 4000 car driver, and 6000 truck drivers. The probability of an accident involving a scooter, a car and a truck are 0.01, 0.03, and 0.15 respectively. One of the insured person meets with an accident. What is the probability that he is a scooter driver?

Sol^{n}:

Let, E1, E2, and E3 be the events of a driver being a scooter driver, car driver, and truck driver respectively.

Let A be the event that the person meets with an accident.

$$P(E1) = \frac{2000}{1200} \quad = \frac{1}{6} \quad = 0.16$$

$$P(E2) = \frac{4000}{1200} \quad = \frac{1}{3} \quad = 0.3$$

$$P(E3) = \frac{6000}{1200} \quad = \frac{1}{2} \quad = 0.5$$

Also, we have,

$P(A|E1) = 0.01$

$P(A|E2) = 0.03$

$P(A|E3) = 0.15$

Now, the probability that the insured person who meets with an accident is a scooter driver is $P(E1|A)$

Using Baye's Theorem;

$$\boldsymbol{P(E1|A) = \frac{P(E1) * P(A|E1)}{P(E1) * P(A|E1) + P(E2) * P(A|E2) + P(E3) * P(A|E3)}}$$

$$\boldsymbol{= \frac{0.16 * 0.01}{0.16 * 0.01 + 0.3 * 0.03 + 0.5 * 0.15}}$$

$$\boldsymbol{= \frac{0.0016}{0.0016 + 0.009 + 0.075}}$$

$$\boldsymbol{= \frac{0.0016}{0.0856}}$$

$$\boldsymbol{= 0.018}$$

Department of Computer Science – DEC [NAST]

Causal Network

Belief networks have often been called **causal networks** and have been claimed to be a good representation of causality. Recall that a causal model predicts the result of interventions.

A **belief network**, also called a **Bayesian network**, is an acyclic directed graph (DAG), where the nodes are random variables. There is an arc from each element of $parents(X_i)$ into X_i. Associated with the belief network is a set of conditional probability distributions - the conditional probability of each variable given its parents (which includes the prior probabilities of those variables with no parents).

Thus, a belief network consists of

- a DAG, where each node is labeled by a random variable;
- a domain for each random variable; and
- a set of conditional probability distributions giving $P(X|parents(X))$ for each variable X.
- set of interconnected nodes, where each node represents a variable in the dependency model and the connecting arcs represent the causal relationship between these variables.

A belief network is acyclic by construction. The way the chain rule decomposes the conjunction gives the ordering. A variable can have only predecessors as parents. Different decompositions can result in different belief networks.

Reasoning in Belief Network

- A Belief Network is also called a Bayesian Network.
- A Bayesian Belief Network is a graphical representation of a probabilistic dependency model.
- Bayesian networks have been the most important contribution to the field of AI in the last 10 years.
- Provide a way to represent knowledge in an uncertain domain and a way to reason about this knowledge.
- Many applications: medicine, factories, help desks, spam filtering, etc.

- A Bayes net (also called a belief network) is an augmented directed acyclic graph, represented by the pair V , E

where:

– **V** is a set of vertices.

– **E** is a set of directed edges joining vertices. No loops of any length are allowed.

- Each vertex in V contains the following information:

– The name of a random variable

– A probability distribution table indicating how the probability of this variable's values depends on all possible combinations of parental values.

Example:

You have a new burglar alarm installed at home. It is fairly reliable at detecting a burglary, but also responds on occasion to minor earthquakes. You also have two neighbors, John and Mary, who have promised to call you at work when they hear the alarm. John always calls when he hears the alarm, but sometimes confuses the telephone ringing with the alarm and calls then, too. Mary, on the other hand, likes rather loud music and sometimes misses the alarm altogether. Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.

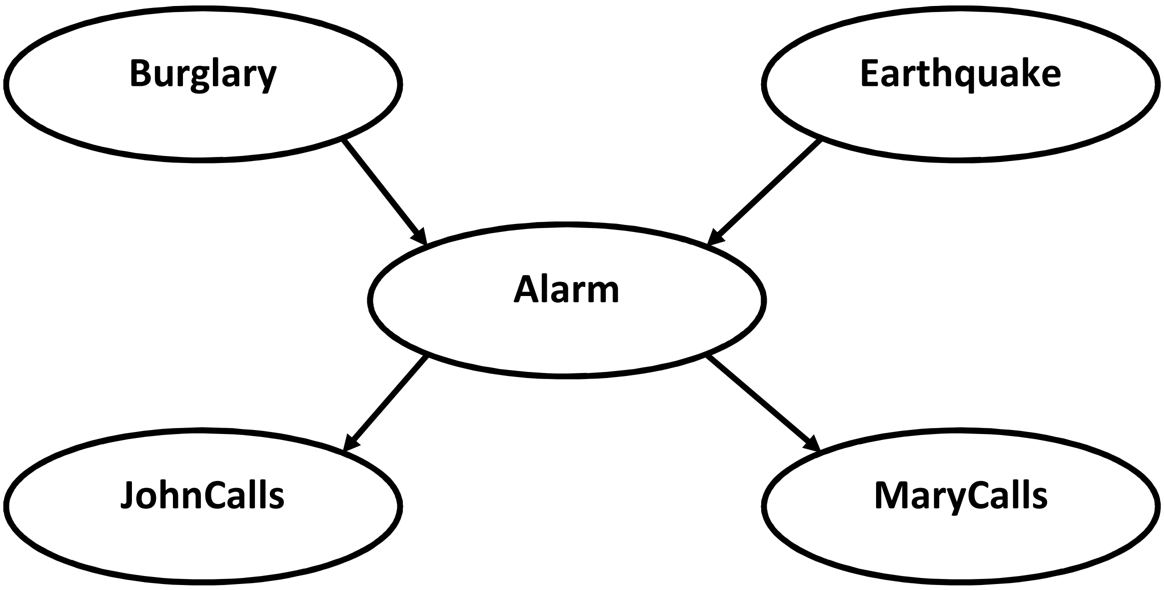


Fig.: A Typical Belief Network

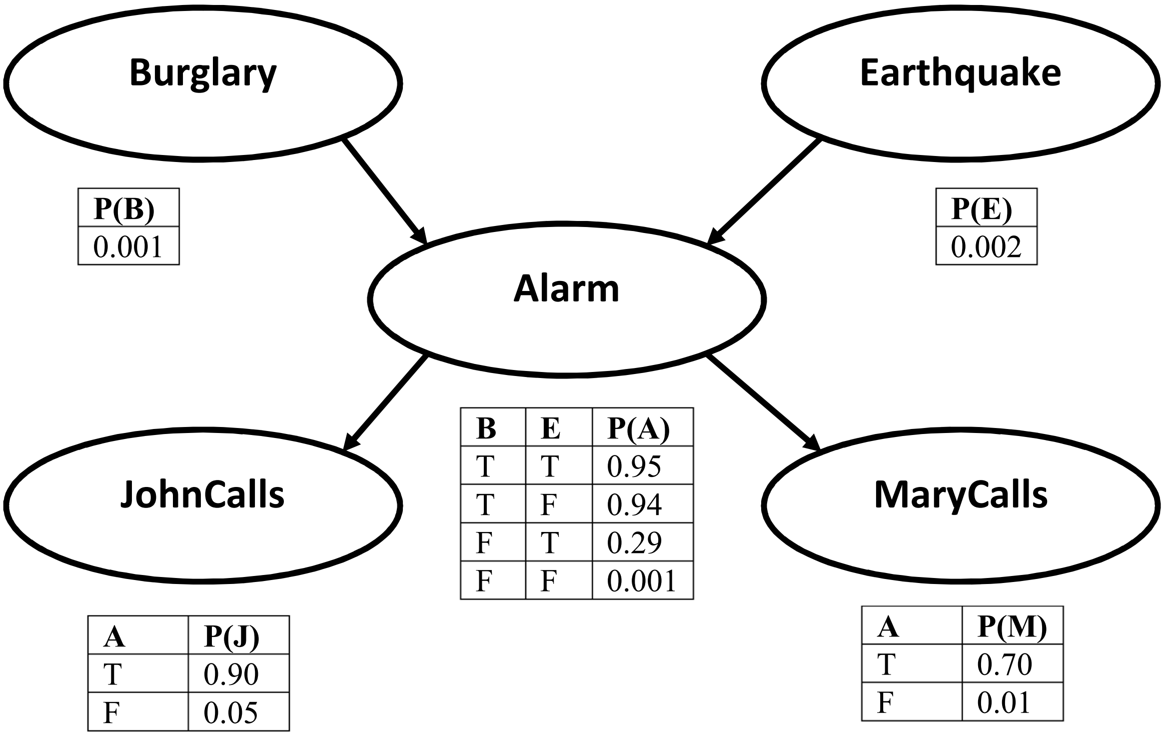
A typical belief network with conditional probabilities. The letters **B**, **E**, **A**, **J**, and **M** stands for **Burglary**, **Earthquake**, **Alarm**, **JohnCalls**, and **MarryCalls**, respectively. All variables (nodes) are Boolean, so the probability of, say **~P(A)** in any row of its table is **1 - P(A)**

P(JohnCalls, MaryCalls, Alarm, Burglary, Earthquake)

= P(JohnCalls | Alarm) P(MaryCalls | Alarm) P(Alarm | Burglary,Earthquake) P(Burglary) P(Earthquake)

OR

P(J, M, A, B, E) = P(J | A) P(M | A) P(A | B, E) P(B) P(E)



P(B)
0.001

P(E)
0.002

B	E	P(A)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(J)
T	0.90
F	0.05

A	P(M)
T	0.70
F	0.01

Q) Calculate the probability of the event that the alarm has sounded but neither a burglary nor an earthquake has occurred and John and Mary call.

Sol^n: $P(J \wedge M \wedge A \neg B \wedge \neg E) = ?$

$$= P(J|A) * P(M|A) * P(A|\neg B, \neg E) * P(\neg B) * P(\neg E)$$

$$= 0.90 * 0.70 * 0.001 * (1 - 0.001) * (1 - 0.002)$$

$$= 0.90 * 0.70 * 0.001 * 0.999 * 0.998$$

$$= \mathbf{0.00062}$$

4.6 Semantic Nets and Frames

Semantic networks are alternative of predicate logic for knowledge representation. In Semantic networks, we can represent our knowledge in the form of graphical networks. This network consists of nodes representing objects and arcs which describe the relationship between those objects. Semantic networks can categorize the object in different forms and can also link those objects. Semantic networks are easy to understand and can be easily extended.

This representation consists of mainly two types of relations:

i. IS-A relation (Inheritance)
ii. Kind-of-relation

Example: Following are some statements which we need to represent in the form of nodes and arcs.

Statements:

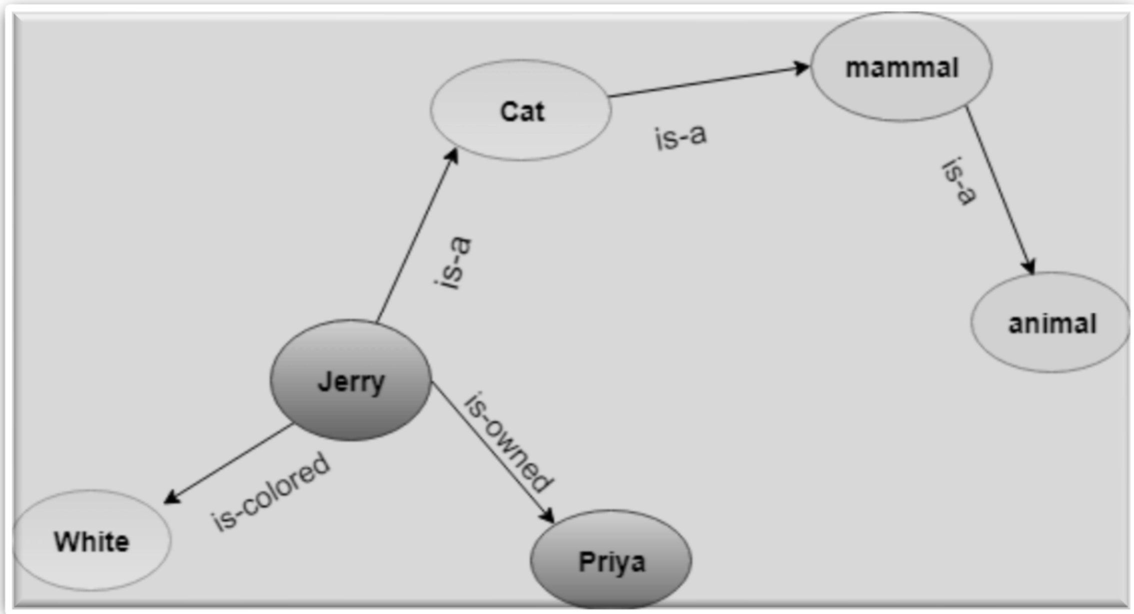
a. Jerry is a cat.

b. Jerry is a mammal

c. Jerry is owned by Priya.

d. Jerry is brown colored.

e. All Mammals are animal.



Department of Computer Science – DEC [NAST]

In the above diagram, we have represented the different type of knowledge in the form of nodes and arcs. Each object is connected with another object by some relation.

Drawbacks in Semantic representation:

1. Semantic networks take more computational time at runtime as we need to traverse the complete network tree to answer some questions. It might be possible in the worst-case scenario that after traversing the entire tree, we find that the solution does not exist in this network.
2. Semantic networks try to model human-like memory (Which has 1015 neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.
3. These types of representations are inadequate as they do not have any equivalent quantifier, **e.g., for all, for some, none, etc.**
4. Semantic networks do not have any standard definition for the link names.
5. These networks are not intelligent and depend on the creator of the system.

Advantages of Semantic network:

1. Semantic networks are a natural representation of knowledge.
2. Semantic networks convey meaning in a transparent manner.
3. These networks are simple and easily understandable.

3. Frame Representation

A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world. Frames are the AI data structure which divides knowledge into substructures by representing stereotypes situations. It consists of a collection of slots and slot values. These slots may be of any type and sizes. Slots have names and values which are called facets.

Facets: The various aspects of a slot are known as **Facets**. Facets are features of frames which enable us to put constraints on the frames. Example: IF-NEEDED facts are called when data of any particular slot is needed. A frame may consist of any number of slots, and a slot may include any number of facets and facets may have any number of values. A frame is also known as **slot-filter knowledge representation** in artificial intelligence.

Frames are derived from semantic networks and later evolved into our modern-day classes and objects. A single frame is not much useful. Frames system consist of a collection of frames which are connected. In the frame, knowledge about an object or event can be stored together in the knowledge base. The frame is a type of technology which is widely used in various applications including Natural language processing and machine visions.

Example: 1

Let's take an example of a frame for a book

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

Example 2:

Let's suppose we are taking an entity, Peter. Peter is an engineer as a profession, and his age is 25, he lives in city London, and the country is England. So, following is the frame representation for this:

Slots	Filter
Name	Peter
Profession	Doctor
Age	25
Marital status	Single
Weight	78

Advantages of frame representation:

1. The frame knowledge representation makes the programming easier by grouping the related data.
2. The frame representation is comparably flexible and used by many applications in AI.
3. It is very easy to add slots for new attribute and relations.
4. It is easy to include default data and to search for missing values.
5. Frame representation is easy to understand and visualize.

Disadvantages of frame representation:

1. In frame system inference mechanism is not be easily processed.
2. Inference mechanism cannot be smoothly proceeded by frame representation.
3. Frame representation has a much-generalized approach.

THE END

Department of Computer Science – DEC [NAST]