# Part 3

**9.3 Knowledge representation**: Knowledge representations and Mappings, Approaches to Knowledge Representation, Issues in Knowledge Representation, Semantic Nets, Frames, Propositional Logic(PL) (Syntax, Semantics, Formal logic-connectives, tautology, validity, well-formed-formula, Inference using Resolution), Predicate Logic (FOPL, Syntax, Semantics, Quantification, Rules of inference, unification, resolution refutation system), Bayes' Rule and its use, Bayesian Networks, and Reasoning in Belief Networks. (ACtE0903)

**Note →We are dealing all the topics w.r.t to the rational agent perspective and hence design the AI systems from the agent's way**

# Why Knowledge Representation??

- **Problem solving through searching does not involve the representation of facts of the world.**
- Ex →Best first search(BFS) just simply generates successors and computes the h(n) function without reference to domain specific knowledge.
- However to solve more real complex world problems we need a lot of
- **1. facts(knowledge) about the world related to the problem**
- **2.** and mechanisms to manipulate these facts .
- **This is where knowledge representation comes in.**

# Knowledge and Knowledge Representation

- Consists of facts, concepts, rules about the world etc and these are represented as pictures, texts or anything that an agent can understand
- Knowledge Representation → express the knowledge about the world in a computer-tractable form. i.e. express the knowledge about the world in such a way that computers can handle it

# knowledge representation

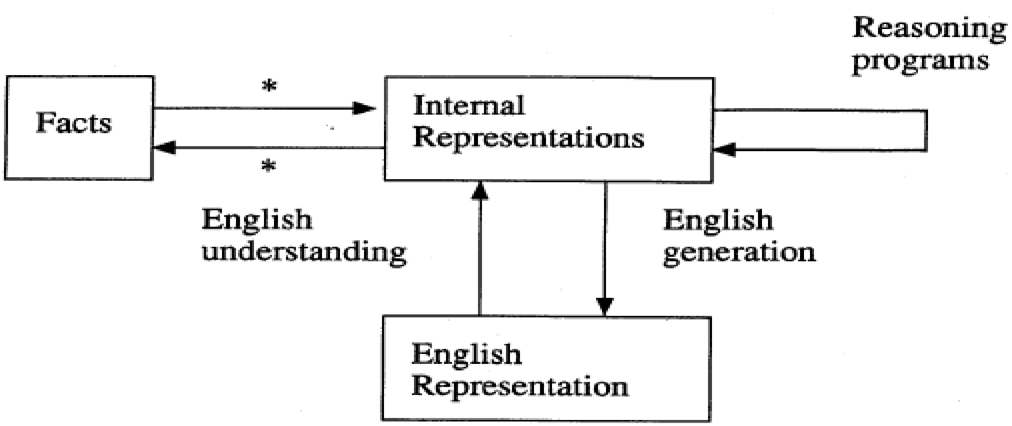
- Key aspects of knowledge representation languages are:
  - **Syntax:** describes how sentences are formed in the language.
  - **Semantics:** describes the meaning of sentences, what is it the sentence refers to in the real world.
  - **Computational aspect:** describes how sentences and objects are manipulated in concordance with semantically conventions.
- A ***formal language*** is required to represent knowledge in a computer tractable form and r*easoning* processes are required to manipulate this knowledge to deduce non-obvious facts.

# Facts and Representations

- Facts →Truths in the world. <u>It is the fact that we want to represent.</u>
- We need to representation facts in some formal/ mathematical way.
- Once we represent the facts only then can we manipulate them

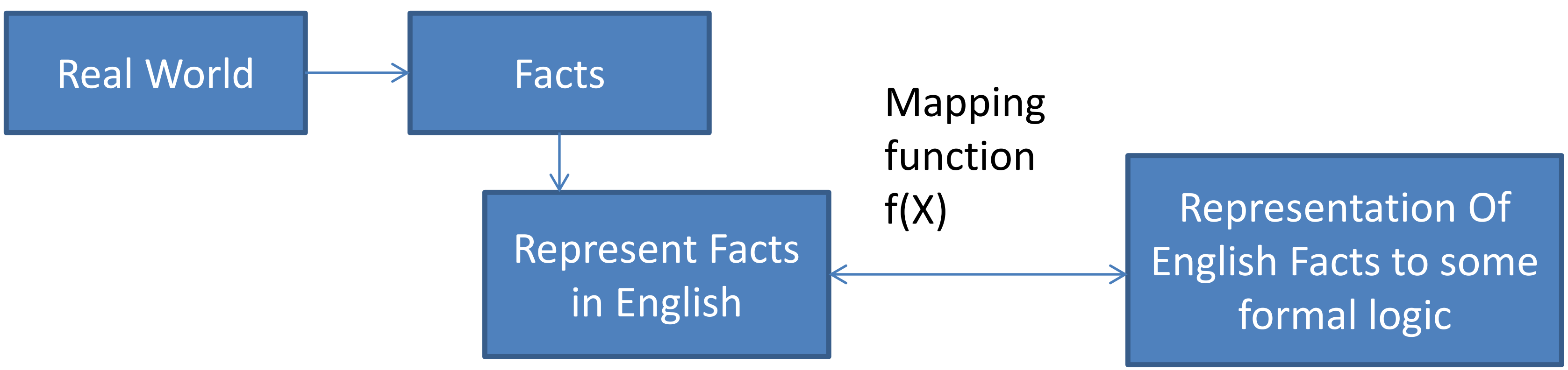
# Mapping Between Facts and Representations

- Mapping is the process that maps facts to representations and vice versa.
- The forward representation mapping maps from facts to representations while the backward representation mapping maps from representations to facts.
  - It shows the relationship between the objects.
- At knowledge level facts are described and **at symbol level facts are defined in terms of symbols so that the symbols can be manipulated by the computer program**



# How do we represent Facts????

- 1. One way to represent fact is using one of the natural language (the English language)
- 2. Now we can have the English representation of those facts to facilitate getting information into and out of the system
- 3. Once we represent the facts in English we can have a mapping function to map the English sentences into the representation that we are actually going to use and vice versa



# An example → Mapping between Facts and Representations

Src: [https://www.pinterest.com/rubabjii/coloured-tail/](https://www.pinterest.com/rubabjii/coloured-tail/)

Fact Represented in English Language→
**Tommy is a Dog**

Mapping function f(X)

Representation of English Language into some logical form→
**Dog(Tommy)**

Lets suppose we also have a logical representation of the fact that :→

**All dogs have tail → $\forall x : Dog(x) \Rightarrow has\_tail(x)$**

Now from the facts : **Dog(Tommy) and $\forall x : Dog(x) \Rightarrow has\_tail(x)$**

we can derive new fact

→ **has_tail(Tommy) --------Mapping to English Language------ > Tommy has tail**

# Knowledge Representation Using Logic

- **Logic** is defined as a formal language for expressing knowledge and ways of reasoning.
- Logic makes statements about the world which are true (or false).
- Logic combines the advantages of natural languages and formal languages.
- Logic is:
  - concise
  - unambiguous
  - context insensitive
  - expressive
  - effective for inferences

# Logic

A logic is defined by the following:

- ***Syntax*** - describes the possible configurations that constitute sentences.
- ***Semantics*** - determines what facts in the world the sentences refer to **i.e. the interpretation. Each sentence makes a claim about the world**.
- ***Proof theory*** - **set of rules for generating new sentences that are necessarily true given that the old sentences are true.** The relationship between sentences is called **entailment**. The semantics link these sentences (representation) to facts of the world. **The proof can be used to determine new facts which follow from the old.**
- ***A set of sentences*** – A sentence is constructed from a set of primitives according to syntax rules.
- ***A set of interpretations*** – An interpretation gives a semantic to primitives. It associates primitives with values.
- ***The valuation (meaning) function*** – **Assigns a value (typically the truth value) to a given sentence under some interpretation. sentence × interpretation → {True , False }**

# Knowledge Representation Using Logic

- **How logic can be used to form representations of the world?** → All men are Mortal:
- $\forall$**x : Man(x)** $\Rightarrow$ **Mortal(x)**

- **How a process of inference can be used to derive new representations about the world**?
- Marcus is a Man : Man(Marcus)
- {Man(Marcus) , $\forall$x : Man(x) $\Rightarrow$ Mortal(x) } **derives Mortal(Marcus)**

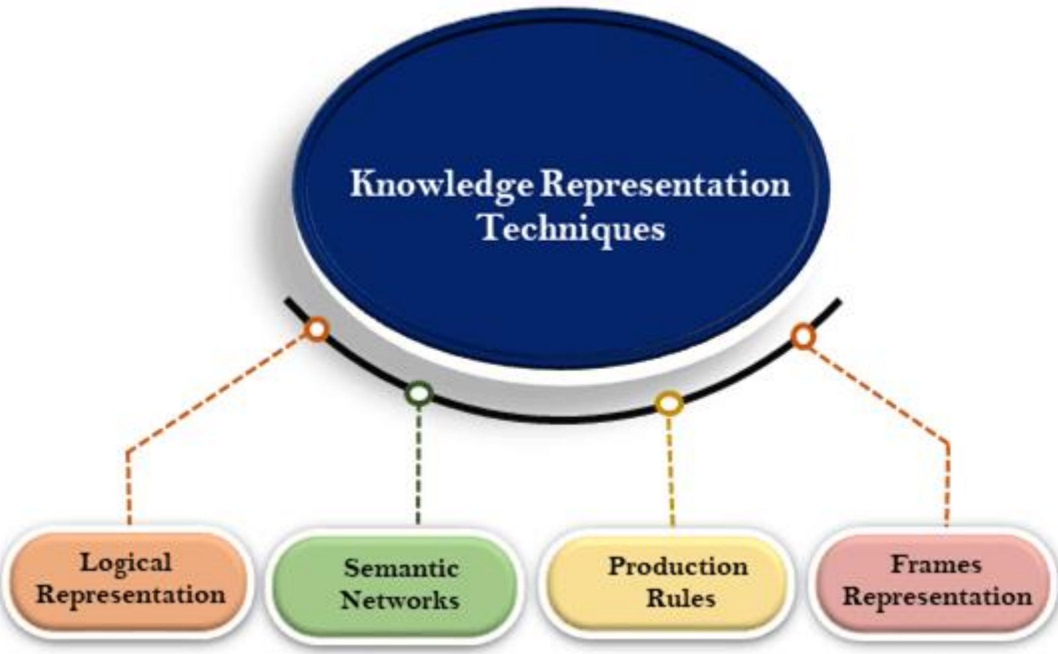
- How these can be used by an intelligent agent to deduce what to do.

# Approaches to Knowledge Representation

1. Representable adequacy → ability to represent all knowledge needed in the domain.

2. Inferential adequacy → ability to manipulate knowledge to drive new structures inferred from old.
3. Inferential efficiency →ability to perform inference in the most efficient directions
d) Acquisitioned efficiency → ability to acquire new information easily.

# Knowledge Representation Techniques



# Knowledge Types

- Declarative Knowledge → Concepts, facts, beliefs (that are either true or false) are represented in the form of logic. Describes what property /expresses facts.
- Also called descriptive knowledge and expressed in declarative sentences.
- simpler than procedural language.
- Example → Kathmandu is capital of Nepal
- = Capital _Nepal (Kathmandu)
- Either Kathmandu is capital of Nepal or Madrid is capital of U.K.

= Capital _Nepal (Kathmandu) $\vee$ Capital _UK (Madrid)

- All Roman are Pompeians: $\forall$**x : Roman(x)** $\Rightarrow$ **is_ Pompeian(x)**

- Procedural/Imperative/Operational Knowledge→ a type of knowledge which is responsible for knowing how to do something.
- **Specifies what to do when**
- **It includes rules, strategies, procedures, agendas, etc**.
- Procedural knowledge depends on the task on which it can be applied.
- **Production rule** [ (condition, action) pairs which mean, "If condition then action")] **is commonly used technique to represent procedural knowledge**
- **Example Procedural Knowledge as Rules**
- **IF (at bus stop AND bus arrives) THEN action (get into the bus)**
- **Another example → Arranging the tiles of sliding block puzzle step by step to reach a specified configuration**

**Heuristic Knowledge→**knowledge related to specific domain

Used to make judgments and simplify solution to a problem.

**They help achieve goals quickly.**

**Example →following the shortest path from city A to city B among many paths between two cities**.

# Issues in Knowledge Representation

- Are there any attributes of object so basic that they occur in almost every problem domain?

- Are there any imp relationships between attributes of objects?

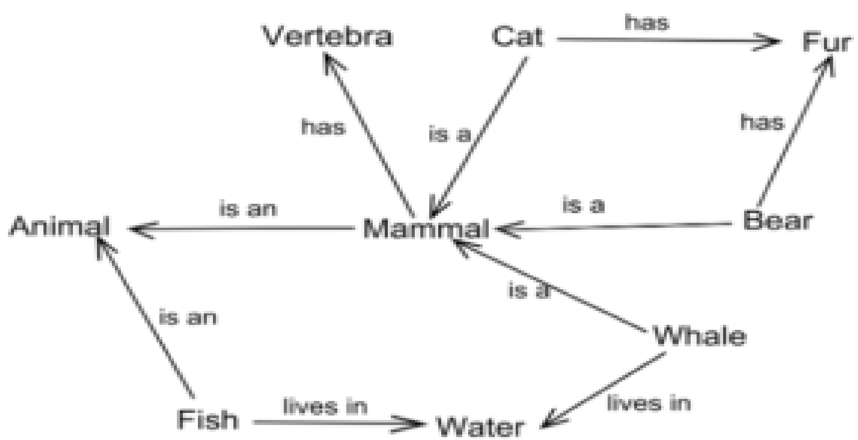
- At what level should knowledge be represented? Is there good set of primitives into which all knowledge can be broken down? Is it helpful to use such primitives?

- How should sets of objects be represented?

- given a large amount of knowledge stored in database ,how can relevant parts be accessed when needed.

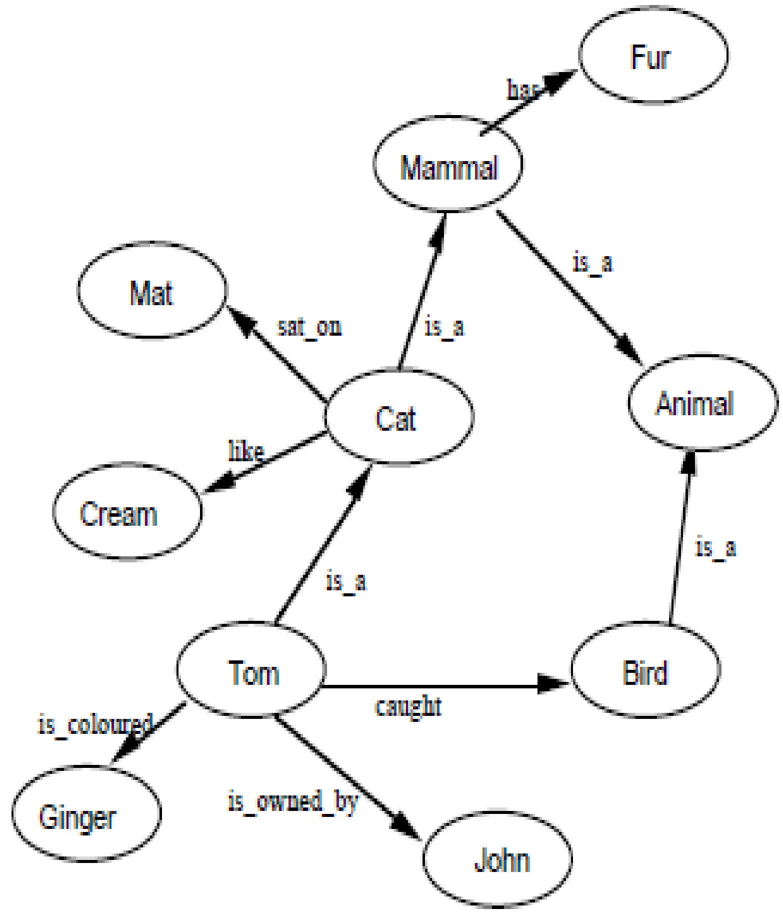
# Semantic Network → Another form of knowledge Representation

- Semantic networks are a **way of representing relationships between objects and ideas.**
- **Also support automated systems for reasoning about the knowledge**
- **In it nodes are objects, events , subjects and arcs are links or relations**
- For example, a Semantic network might tell a computer **the relationship between different animals** (a cat IS A mammal, a cat HAS whiskers).



**Below is an example of a semantic network for the following statements:**

- Tom is a cat.
- Tom caught a bird.
- Tom is owned by John.
- Tom is ginger in colour.
- Cats like cream.
- The cat sat on the mat.
- A cat is a mammal.
- A bird is an animal.
- All mammals are animals.
- Mammals have fur



# Semantic Network → Advantages and Disadvantages

- **Advantages of Semantic network:**
- Semantic networks are a **natural representation of knowledge**.
- Semantic networks **convey meaning in a transparent manner**.
- These networks are **simple and easily understandable.**

## Drawbacks in Semantic representation:

- **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.
- 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.
- **These types of representations are inadequate as they do not have any equivalent quantifier, e.g., for all, for some, none, etc.**
- Semantic networks do not have any standard definition for the link names.
- These networks are not intelligent and depend on the creator of the system.

# Frames

- **a record like structure which consists of a collection of attributes and its values to describe an entity in the world.**
- Is similar to field-value structure corresponding slots and slot fillers
  - It is a **collection of slots and fillers that defines an objects**
  - Frame provides a third dimension representation of knowledge to semantic nets by **allowing nodes to have structures**
- **Represent the given knowledge in frame →**
- 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 |
| Age | 25 |
| Profession | Engineer |
| City | London |
| Country | England |

# Frames

- **Frames system consist of a collection of frames which are connected to each other.**
- **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.

| Cat | |
|---|---|
| likes | bird |
| likes | ice cream |
| type | animal |

is-a

| Tom | |
|---|---|
| like | bird |
| like | ice cream |
| type | animal |
| type | Ginger |
| like | John |

part_of

| John | |
|---|---|
| Age | 25 |
| Color | White |

# Frames Advantages and Disadvantages

- Advantages of frame representation:

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

Disadvantages of frame representation:

- Inference mechanism cannot be smoothly proceeded by frame representation.
- Frame representation has a much generalized approach.

# Propositional Logic

- A propositional logic is a **declarative** sentence which can be either true or false but not both or either.

All the following declarative sentences are propositions.

1. Washington, D.C., is the capital of the United States of America.
2. Toronto is the capital of Canada.
3. $1 + 1 = 2$.
4. $2 + 2 = 3$.

Propositions 1 and 3 are true, whereas 2 and 4 are false.

Consider the following sentences.

1. What time is it?
2. Read this carefully.
3. $x + 1 = 2$.
4. $x + y = z$.

Sentences 1 and 2 are not propositions because they are not declarative sentences. Sentences 3 and 4 are not propositions because they are neither true nor false. Note that each of sentences 3 and 4 can be turned into a proposition if we assign values to the variables.

# Propositional Logic

- Propositional logic is **a mathematical model that allows us to reason about the truth or false of logical expression**.
- In **propositional logic, there are atomic sentences** and compound sentences (built up from atomic sentences using logical connectives)

# Propositional Logic

- **Logical constants**: true, false
- **Propositional symbols**: P, Q, S, ... (**atomic sentences**)
- Wrapping **parentheses**: ( … )
- Sentences are combined by **logical connectives**:

| | | |
|---|---|---|
| $\wedge$ | ...and | [conjunction] |
| $\vee$ | ...or | [disjunction] |
| $\Rightarrow$ | ...implies | [implication / conditional] |
| $\Leftrightarrow$ | ..is equivalent | [biconditional] |
| $\neg$ | ...not | [negation] |

- **Literal**: atomic sentence or negated atomic sentence

# Propositional Logic

- A simple language useful for showing key ideas and definitions
- User defines a set of propositional symbols, like P and Q.
- User defines the **semantics** of each propositional symbol:
  - P means “It is hot”
  - Q means “It is humid”
  - R means “It is raining”

- Examples of PL sentences

- P = “it is hot” and Q = “It is humid.”
- $(P \wedge Q) \rightarrow R$
  “If it is hot and humid, then it is raining”
- $Q \rightarrow P$
  “If it is humid, then it is hot”

# Well formed formula

- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg S$ is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then $(S \vee T)$, $(S \wedge T)$, $(S \rightarrow T)$, and $(S \leftrightarrow T)$ are sentences
  - A sentence results from a finite number of applications of the above rules

# Additional Terms

- The meaning or **semantics** of a sentence determines its **interpretation**.
- Given the truth values of all symbols in a sentence, it can be "evaluated" to determine its **truth value** (True or False).
- A **model** for a KB is a "possible world" (assignment of truth values to propositional symbols) in which each sentence in the KB is True.

# Additional Terms

- A **valid sentence** or **tautology** is a sentence that is True under all interpretations, no matter what the world is actually like or what the semantics is. Example: "It's raining or it's not raining."
- An **inconsistent sentence** or **contradiction** is a sentence that is False under all interpretations. The world is never like what it describes, as in "It's raining and it's not raining."
- A compound proposition that is neither a tautology nor a contradiction is called a ***contingency***.
- In below figure Because $p \vee \neg p$ is always true, it is a tautology. Because $p \wedge \neg p$ is always false, it is a contradiction.

Examples of a Tautology and a Contradiction.

| $p$ | $\neg p$ | $p \vee \neg p$ | $p \wedge \neg p$ |
|---|---|---|---|
| T | F | T | F |
| F | T | T | F |

- **P entails Q**, written **P |= Q**, means that whenever P is True, so is Q. In other words, all models of P are also models of Q.

# Logical Equivalence

Show that $p \rightarrow q$ and $\neg p \vee q$ are logically equivalent.

*Solution:* We construct the truth table for these compound propositions in Table 4. Because the truth values of $\neg p \vee q$ and $p \rightarrow q$ agree, they are logically equivalent. ◀

**TABLE 4** **Truth Tables for $\neg p \vee q$ and $p \rightarrow q$.**

| $p$ | $q$ | $\neg p$ | $\neg p \vee q$ | $p \rightarrow q$ |
|---|---|---|---|---|
| T | T | F | T | T |
| T | F | F | F | F |
| F | T | T | T | T |
| F | F | T | T | T |

# Truth Table for Propositional Logic

$\wedge$ ...and [conjunction]

$\vee$ ...or [disjunction]

$\Rightarrow$...implies [implication / conditional]

$\Leftrightarrow$..is equivalent [biconditional]

$\neg$ ...not [negation

| A | B | ¬ A | A V B | A ∧ B | A => B | A ⇔B |
|---|---|---|---|---|---|---|
| T | T | F | T | T | T | T |
| T | F | F | T | F | F | F |
| F | T | T | T | F | T | F |
| F | F | T | F | F | T | T |

# CONVERSE, CONTRAPOSITIVE, AND INVERSE

## The variations of Conditional Statements

- **Given a conditional statement $p \rightarrow q$**
- The proposition $q \rightarrow p$ is called the **converse** of $p \rightarrow q$.
- The **contrapositive** of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$.
- The proposition $\neg p \rightarrow \neg q$ is called the **inverse** of $p \rightarrow q$.
- Out of these three conditional statements formed from $p \rightarrow q$, only the contrapositive always has the same truth value as $p \rightarrow q$.

# Note p→q can also be denoted in below terms

| | |
|---|---|
| "if $p$, then $q$" | "$p$ implies $q$" |
| "if $p$, $q$" | "$p$ only if $q$" |
| "$p$ is sufficient for $q$" | "a sufficient condition for $q$ is $p$" |
| "$q$ if $p$" | "$q$ whenever $p$" |
| "$q$ when $p$" | "$q$ is necessary for $p$" |
| "a necessary condition for $p$ is $q$" | "$q$ follows from $p$" |
| "$q$ unless $\neg p$" | |

# Converse, inverse and Contrapositive

What are the contrapositive, the converse, and the inverse of the conditional statement

"The home team wins whenever it is raining?"

*Solution:* Because "$q$ whenever $p$" is one of the ways to express the conditional statement $p \rightarrow q$, the original statement can be rewritten as

"If it is raining, then the home team wins."

Consequently, the contrapositive of this conditional statement is

"If the home team does not win, then it is not raining."

The converse is

"If the home team wins, then it is raining."

The inverse is

"If it is not raining, then the home team does not win."

Only the contrapositive is equivalent to the original statement.

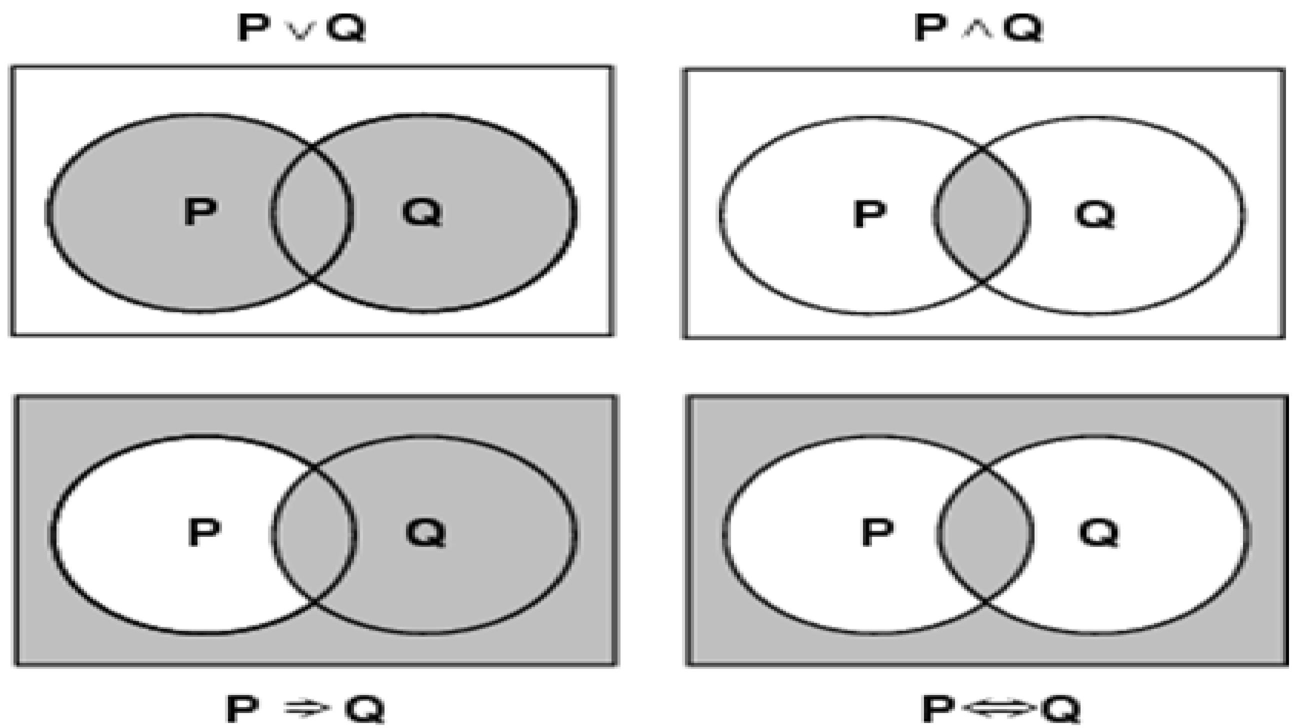
# Complex Sentences

A complex sentence:

| $P$ | $H$ | $P \vee H$ | $(P \vee H) \wedge \neg H$ | $((P \vee H) \wedge \neg H) \Rightarrow P$ |
|---|---|---|---|---|
| *False* | *False* | *False* | *False* | *True* |
| *False* | *True* | *True* | *False* | *True* |
| *True* | *False* | *True* | *True* | *True* |
| *True* | *True* | *True* | *False* | *True* |

**It's important to know these relations in diagram and make sense with sentences involving ALL, NONE, SOME, IF conditional and IFF (Biconditonal) sentences**

# Models of complex sentences



# Inference

- Inference –Deriving new sentence from the old
- In it when one asks a question of the knowledge base, **the answer should follow from what has been told to the knowledge base previously**
- i.e. one sentence follows from the sentences of the knowledge. And hence we derive a new sentence.
- **Logical inference** is used to create new sentences that logically follow from a given set of predicate calculus sentences (KB).

# Inference Soundness and Completeness

- An inference rule is **sound** if every sentence X produced by an inference rule operating on a KB logically follows from the KB. (That is, the inference rule does not create any contradictions)
- An inference rule is **complete** if it is able to produce every expression that logically follows from (is entailed by) the KB. (Note the analogy to complete search algorithms.)

# Syllogism

- A syllogism, (from the Greek words for conclusion and inference,) is a logic puzzle **where you draw a conclusion** →**from particular kinds of purported facts you are given(knowledge Base )** and those facts alone.
- Syllogisms are an important basis of logical thinking.

Src:https://teachinglondoncomputing.org/sherlock-syllogisms/

# Example →Syllogism

**All gems in the game are expensive in-game purchases.**

**All rubies in the game are gems.**

- Therefore which of the following can we conclude?
- a. Some rubies in the game are expensive in-game purchases.
- b. All rubies in the game are expensive in-game purchases.
- c. Some gems in the game are expensive in-game purchases.
- d. None of the above.

# Rules OF Inference

| Rule of Inference | Tautology | Name |
|---|---|---|
| $p$<br>$p \to q$<br>$\therefore q$ | $(p \wedge (p \to q)) \to q$ | Modus ponens |
| $\neg q$<br>$p \to q$<br>$\therefore \neg p$ | $(\neg q \wedge (p \to q)) \to \neg p$ | Modus tollens |
| $p \to q$<br>$q \to r$<br>$\therefore p \to r$ | $((p \to q) \wedge (q \to r)) \to (p \to r)$ | Hypothetical syllogism |
| $p \vee q$<br>$\neg p$<br>$\therefore q$ | $((p \vee q) \wedge \neg p) \to q$ | Disjunctive syllogism |
| $p$<br>$\therefore p \vee q$ | $p \to (p \vee q)$ | Addition |
| $p \wedge q$<br>$\therefore p$ | $(p \wedge q) \to p$ | Simplification |
| $p$<br>$q$<br>$\therefore p \wedge q$ | $((p) \wedge (q)) \to (p \wedge q)$ | Conjunction |
| $p \vee q$<br>$\neg p \vee r$<br>$\therefore q \vee r$ | $((p \vee q) \wedge (\neg p \vee r)) \to (q \vee r)$ | Resolution |

State which rule of inference is the basis of the following argument: "It is below freezing now. Therefore, it is either below freezing or raining now."

*Solution:* Let $p$ be the proposition "It is below freezing now" and $q$ the proposition "It is raining now." Then this argument is of the form

$$\begin{array}{rl} & p \\ \hline \therefore & p \vee q \end{array}$$

This is an argument that uses the addition rule.

State which rule of inference is the basis of the following argument: "It is below freezing and raining now. Therefore, it is below freezing now."

*Solution:* Let $p$ be the proposition "It is below freezing now," and let $q$ be the proposition "It is raining now." This argument is of the form

$$\begin{array}{rl} & p \wedge q \\ \hline \therefore & p \end{array}$$

This argument uses the simplification rule.

State which rule of inference is used in the argument:

> If it rains today, then we will not have a barbecue today. If we do not have a barbecue today, then we will have a barbecue tomorrow. Therefore, if it rains today, then we will have a barbecue tomorrow.

*Solution:* Let $p$ be the proposition "It is raining today," let $q$ be the proposition "We will not have a barbecue today," and let $r$ be the proposition "We will have a barbecue tomorrow." Then this argument is of the form

$$\begin{array}{rl} & p \rightarrow q \\ & q \rightarrow r \\ \hline \therefore & p \rightarrow r \end{array}$$

Hence, this argument is a hypothetical syllogism.

1. A. Knowledge base (KB) consists of set of statements.
B. Inference is deriving a new sentence from the KB.
Choose the correct option.
a) A is true, B is true
b) A is false, B is false
c) A is true, B is false
d) A is false, B is true

2. Which is not a property of representation of knowledge?
a) Inferential Efficiency
b) Inferential Adequacy
c) Representational Adequacy
d) Representational Verification

3. Inference algorithm is complete only if ______________
a) It can derive any sentence
b) It can derive any sentence that is an entailed version
c) It is truth preserving
d) It can derive any sentence that is an entailed version & It is truth preserving

4. All of the following is true about logic except:

a) Concise

b) Unambiguous

c) context insensitive

d) Expressive

e) effective for inferences

f) None

5. Which of the following is false?

a) “All Roman are Pompeians” is declarative knowledge.

b) If I feel cold then I get a glass of hot water is procedural knowledge.

c) Getting from city A to city B with some pre determined information is not heuristics.

d) None

6. ‘$\alpha \models \beta$ ‘(to mean that the sentence $\alpha$ entails the sentence $\beta$) if and only if, in every model in which $\alpha$ is _____ $\beta$ is also _____

a) True, true

b) True, false

c) False, true

d) False, false

7. Choose the correct option:

a) A *formal language* is required to represent knowledge in a computer tractable form and r*easoning* processes are required to manipulate this knowledge to deduce non-obvious facts.

b) Semantics describes how sentences are formed in the language and Syntax describes the meaning of sentences, what is it the sentence refers to in the real world.

c) Syntax describes how sentences are formed in the language and Semantics describes the meaning of sentences, what is it the sentence refers to in the real world.

d) a and c

# CNF

**Conjunctive normal form(CNF)**: A sentence is written in conjunctive normal form look likes:…………

$(A \vee B \neg C) \wedge (B \vee D) \wedge (\neg A) \wedge (B \vee C)$

$(A \vee B \neg C)$ is a clause

Its outermost structure is conjunction. It's a conjunction of multiple units.these unit called "**clauses**"

**A clause is the disjunction of many things. The unit that make up a clause are called literals.** A literal is either a variable or negation of variable .

**Ex.** $A, B, \neg C$ are literal

**You can take any sentence in propositional logic write it in conjunctive normal form(CNF).**

## 3 Disjunctive and Conjunctive Normal Forms

The rules given in the previous section (such as De Morgan's) allow us to transform propositional formulas (expressions) to equivalent formulas. In this lecture, we will see two "standard forms" that can represent every propositional formula.

**Definition 1.** *A propositional formula is in* disjunctive normal form (DNF) *if it is the disjunction of conjunctions of literals.*

On its own, this definition may seem somewhat cryptic, so we explain each term below:

- A *literal* is a Boolean variable, or the negation of a Boolean variable (e.g., $P$, $\neg P$).
- A *conjunction* is a logical formula that is the AND ($\wedge$) of (smaller) formulas (e.g., $P \wedge \neg Q \wedge R$).
- A *disjunction* is a logical formula that is the OR ($\vee$) of (smaller) formulas (e.g., $P \vee \neg Q \vee R$).

With this in mind, the meaning of DNF should be clearer: a DNF formula is an OR of AND's. For example,

$$X = (A \wedge \neg B \wedge \neg D) \vee (B \wedge C) \vee (C \wedge \neg D \wedge E)$$

is a DNF formula: the literals of $X$ are $A, \neg B, \neg D, B, C, \neg D$, and $E$; the conjunctions are $A \wedge \neg B \wedge \neg D$, $B \wedge C$, and $C \wedge \neg D \wedge E$; and $X$ is the disjunction (OR) of these three conjunctions. On the other hand,

$$Y = (A \vee \neg B) \wedge (B \vee C \vee \neg D)$$

is not a DNF formula, because the structure of the operators is inverted. But this formula *is* in the other "standard" form (CNF) for propositional formulas.

**Definition 2.** *A propositional formula is in* conjunctive normal form (CNF) *if it is the conjunction of disjunctions of literals.*

As noted above, $Y$ is a CNF formula because it is an AND of OR's. The literals of $Y$ are $A, \neg B, B, C$, and $\neg D$; the disjunctions are $(A \vee \neg B)$ and $(B \vee C \vee \neg D)$; and $Y$ is the conjunction (AND) of the two disjunctions.

**Convert to CNF:**

**1. Eliminate arrows using definitions..**

**A→B** = $\neg A \vee B$

**1. Drive negation using de Morgan's Law**

$\neg(A \vee B) = \neg A \wedge \neg B$

$\neg(A \wedge B) = \neg A \vee \neg B$

**3. Distribute OR over AND**

$A \vee (B \wedge C) \Rightarrow (A \vee B) \wedge (A \vee C)$

**Every sentence can be converted to CNF, but it may grow exponentially in size.**

**Ex.**

**(A v B) →( C →D)**

$\neg(A \vee B) \vee (\neg C \vee D)$

$(\neg A \wedge \neg B) \vee (\neg C \vee D)$

$(\neg A \vee \neg C \vee D) \wedge (\neg B \vee \neg C \vee D)$ ……………..CNF good

# Resolution

- Resolution technique uses proof by contradiction and is based on the fact that any sentence in propositional logic can be transformed into an equivalent sentence in conjunctive normal form .The resolution rule yields a sound and complete algorithm for deciding the *satisfiability* of a propositional formula
- The resolution procedure is a simple iterative procedure: at each step , two clauses, called the parent clauses, are compared(resolved), yielding a new clause that has been inferred from them The new clause represent that two parent clauses interact with each other.

**Resolution: Propositional Form**

Once sentences are in clausal form, we can apply the resolution inference process.

One rule is enough.

Given two clauses, this process is:

1.Find two complementary terms (e.g., A and ¬ A) in the two clauses.

2. cancel them

3. form a new clause containing all the remaining terms.

e.g. resolving

X v Y v Z

¬ X v A

---

Y v Z v A

which is a valid deduction.

## Conjunctive Normal Form(CNF)

In propositional logic, the resolution method is applied only to those clauses which are disjunction of literals. **There are following steps used to convert into CNF:**

1) Eliminate bi-conditional implication by replacing A ⇔ B with (A → B) ∧ (B →A)

2) Eliminate implication by replacing A → B with ¬A V B.

3) In CNF, negation(¬) appears only in literals, therefore we move it inwards as:

- **¬ ( ¬A) ≡ A** (double-negation elimination
- **¬ (A ∧ B) ≡ ( ¬A V ¬B)** (De Morgan)
- **¬(A V B) ≡ ( ¬A ∧ ¬B)** (De Morgan)

4) Finally, using distributive law on the sentences, and form the CNF as:

**$(A_1 \vee B_1) \wedge (A_2 \vee B_2) \wedge \ldots \wedge (A_n \vee B_n)$.**

**Note: CNF can also be described as AND of ORS**

Steps for Resolution:
Conversion of facts into propositional logic
Convert propsitional statements into CNF if needed
Negate the statement which needs to prove (proof by contradiction)
Draw resolution graph (unification)

Consider the following Knowledge Base:

1. The humidity is high or the sky is cloudy.
2. If the sky is cloudy, then it will rain.
3. If the humidity is high, then it is hot.
4. It is not hot.

**Goal:** It will rain.

Use propositional logic and apply resolution method to prove that the goal is derivable from the given knowledge base.

**Solution:** Let's construct propositions of the given sentences one by one:

1. Let, P: Humidity is high.

Q: Sky is cloudy.

It will be represented as **P V Q.**

2) Q: Sky is cloudy. ...**from(1)**

Let, R: It will rain.

It will be represented as b**Q → R.**

3) P: Humidity is high. ...**from(1)**

Let, S: It is hot.

It will be represented as **P → S.**

4) ¬S: It is not hot.

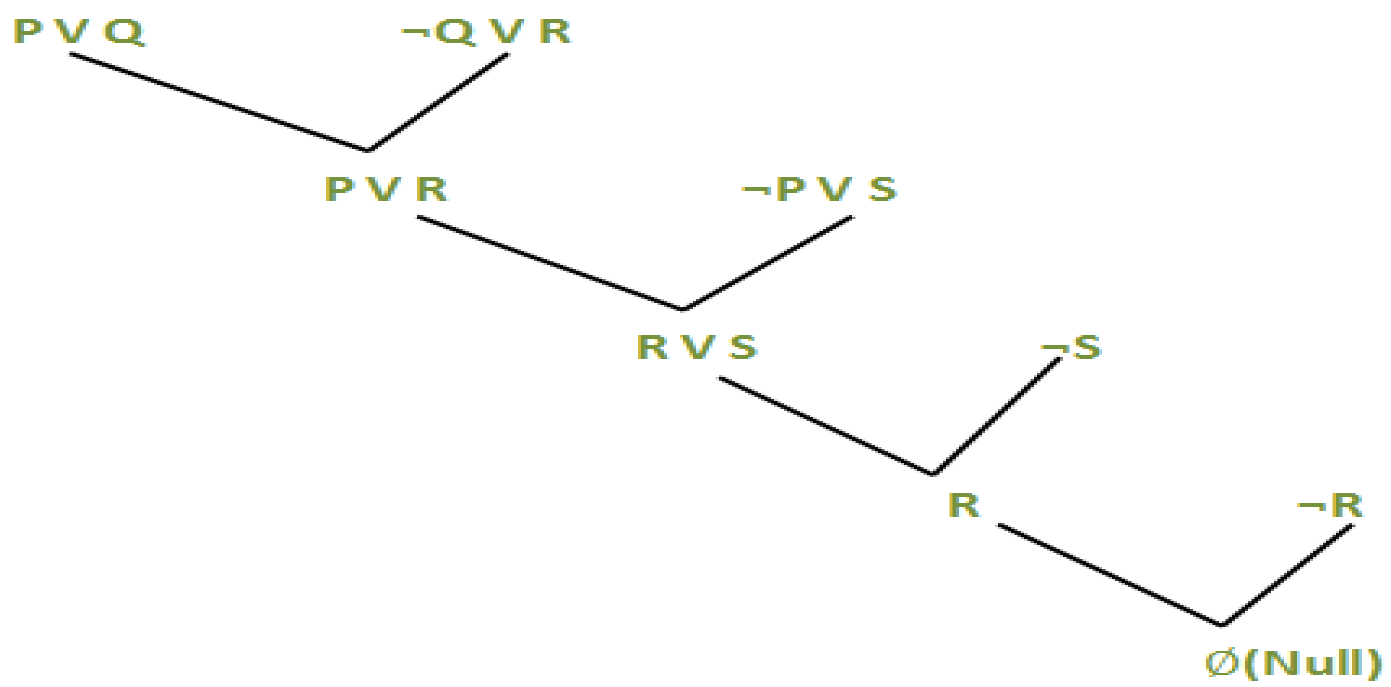
## Applying resolution method:

In (2), Q ⟶ R will be converted as (¬Q V R)

In (3), P ⟶  S will be converted as (¬P V S)

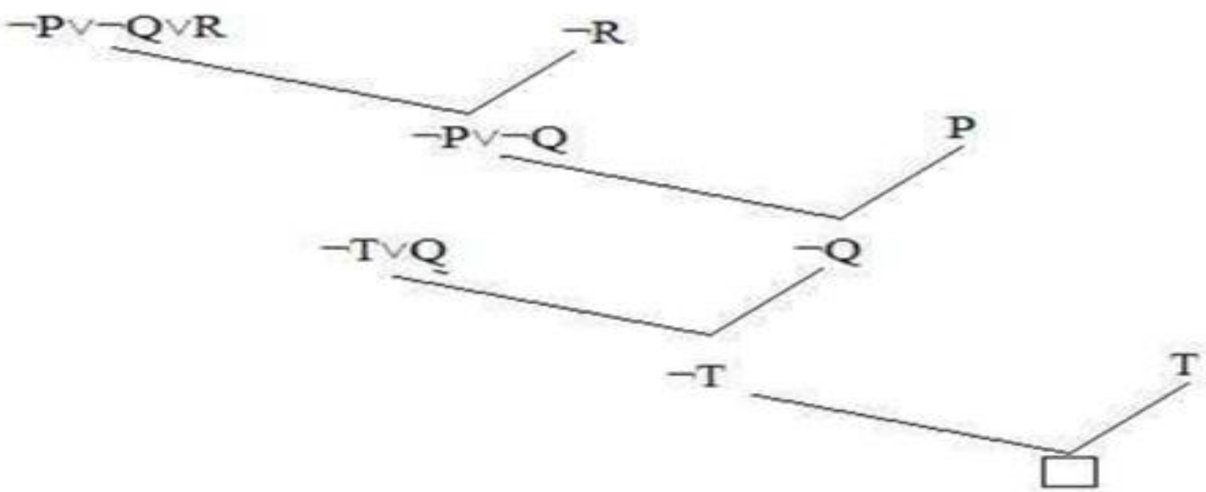
**Negation of Goal (¬R):** It will not rain.

**Finally, apply the rule as shown below:**



# Resolution

| Sl. No | Given Axioms | Converted to Clause Form |
| --- | --- | --- |
| 1 | $P$ | $P$ |
| 2 | $(P \wedge Q) \rightarrow R$ | $\neg P \vee Q \vee R$ |
| 3 | $(S \vee T) \rightarrow Q$ | $\neg S \vee Q$ |
| 4 | | $\neg T \vee Q$ |
| 5 | $T$ | $T$ |



# Problems with Propositional Logic

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

  FOL adds relations, variables, and quantifiers, e.g.,
  - *"Every elephant is gray":* $\forall$ x (elephant(x) $\rightarrow$ gray(x))
  - *"There is a white alligator":* $\exists$ x (alligator(X) ^ white(X))

# First-order logic (FOL)

- First-order logic (FOL) models the world in terms of
  - **Objects,** which are things with individual identities
  - **Properties** of objects that distinguish them from other objects
  - **Relations** that hold among sets of objects
  - **Functions,** which are a subset of relations where there is only one “value” for any given “input”
- Examples:
  - Objects: Students, lectures, companies, cars ...
  - Relations: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
  - Properties: blue, oval, even, large, ...
  - Functions: father-of, best-friend, second-half, one-more-than ...

# FOL Syntax: Basic

- A **term** is used to denote an object in the world
  - **constant**: `BobSmith, 2, Madison, Green, ...`
  - **variable**: `x, y, a, b, c, ...`
  - **function**($term_1, \ldots, term_n$):
    e.g., **`Sqrt(9)`**, **`Distance(Madison, Milwaukee)`**
    - is a relation for which there is one answer
    - maps one or more objects to another *single object*
    - can be used to refer to an unnamed object:
      e.g., **`LeftLegOf(John)`**
    - represents a user-defined *functional* relation
    - cannot be used with logical connectives
- A **ground term** is a term with no variables

- An **atom** is smallest expression
  to which a truth value can be assigned
  - **predicate**($term_1, \ldots, term_n$):
    e.g., **`Teacher(John, Deb)`**, **$\leq$`(Sqrt(2), Sqrt(7))`**
    - is a relation for which there may be more than one answer
    - maps one or more objects to a *truth value*
    - represents a user defined relation
  - $term_1 = term_2$:
    e.g., **`Income(John) = 20K`**, **`1 = 2`**
    - represents the ***equality*** relation when two terms refer to the same object
    - is a predicate in prefix form: $=(term_1, term_2)$

# FOL Syntax: Basic

- A sentence represents a fact in the world that is assigned a truth value
  - atom
  - complex sentence using connectives: $\wedge \vee \neg \Rightarrow \Leftrightarrow$
    e.g., **Friend(Deb,Jim) ⇒ Friend(Jim,Deb)**
    e.g., **>(11,22) ∧ <(22,33)**
  - complex sentence using quantified variables: $\forall$ $\exists$

# FOL Semantics: Assigning Truth

- The atom predicate($term_1$, ..., $term_n$) is true iff the objects referred to by $term_1$, ..., $term_n$ are in the relation referred to by the predicate
- What is the truth value for **F(D, J)**?
  - model:
    **objects: Deb, Jim, Sue, Bob**
    **relation: Friend {<Deb,Sue>,<Sue,Deb>}**
  - interpretation:
    **D means Deb, J means Jim, S means Sue, B means Bob**
    **F($term_1$,$term_2$) means $term_1$ is friend of $term_2$**

# FOL Syntax: Quantifiers

**Universal quantifier**: ∀<variable> <sentence>

- Means the sentence is true **for all** values of *x* in the domain of variable *x*

- Main connective typically ⟹ forming if-then rules
  - *All humans are mammals* becomes in FOL
    $\forall x\ \text{Human}(x) \Rightarrow \text{Mammal}(x)$
    i.e., for all *x*, if *x* is a human then *x* is a mammal
  - **Mammals must have fur** becomes in FOL
    $\forall x\ \text{Mammal}(x) \Rightarrow \text{HasFur}(x)$
    for all *x*, if *x* is a mammal then *x* has fur

$$\forall x\ (\text{Human}(x) \Rightarrow \text{Mammal}(x))$$

- Equivalent to the conjunction of instantiations of *x*:

$$(\text{Human}(\text{Jim}) \Rightarrow \text{Mammal}(\text{Jim})) \wedge$$
$$(\text{Human}(\text{Deb}) \Rightarrow \text{Mammal}(\text{Deb})) \wedge$$
$$(\text{Human}(22) \Rightarrow \text{Mammal}(22)) \wedge \ldots$$

All humans are mammals :

$\forall x\ (Human(x) \rightarrow mammal(x))$ correct form

**not** $\forall x\ (Human(x) \wedge mammal(x))$ incorrect form as described below

# FOL Syntax: Quantifiers

- Common mistake is to use $\wedge$ as main connective
  - results in a blanket statement about *everything*

- For example: $\forall x\ (Human(x) \wedge Mammal(x))$
  - **(Human(Jim) $\wedge$ Mammal(Jim))** $\wedge$
    **(Human(Deb) $\wedge$ Mammal(Deb))** $\wedge$
    **(Human(22) $\wedge$ Mammal(22) )** $\wedge$ …

  - means everything is human and a mammal

# FOL Syntax: Quantifiers

**Existential quantifier:** ∃<variable> <sentence>

- Means the sentence is true
  **for some** value of *x* in the domain of variable *x*

- Main connective is typically ∧
  - **Some humans are old** **becomes in FOL**
  - **∃x Human(x) ∧ Old(x)**
    there exist an *x* such that *x* is a human and *x* is old
  - **Mammals may have arms.** **becomes in FOL**
  - **∃x Mammal(x) ∧ HasArms(x)**
    there exist an x such that x is a mammal and x has arms

∃x (Human(x) ∧ Old(x))

- Equivalent to the disjunction of instantiations of *x*:

(Human(Jim) ∧ Old(Jim)) ∨
(Human(Deb) ∧ Old(Deb)) ∨
(Human(22) ∧ Old(22) ) ∨ ...

Some humans are old.

$\exists x(Human(x) \wedge old(x))$ *correct form*

$\exists x(Human(x)\ ) \rightarrow old(x))$ ***in correct form as described below***

Common mistake is to use $\Rightarrow$ as main connective

– results in a weak statement

For example: $\exists x\ (Human(x) \Rightarrow Old(x))$

– **(Human(Jim) $\Rightarrow$ Old(Jim))** $\vee$
  **(Human(Deb) $\Rightarrow$ Old(Deb))** $\vee$
  **(Human(22) $\Rightarrow$ Old(22) )** $\vee$ …

– true if there is *anything* that *isn't* human

# FOL Syntax: Quantifiers

- Properties of quantifiers:
  - $\forall x \, \forall y$ is the same as $\forall y \, \forall x$
  - $\exists x \, \exists y$ is the same as $\exists y \, \exists x$
  - note: $\exists x \, \exists y$ can be written as $\exists x,y$ likewise with $\forall$

- Examples
  - $\forall x \, \forall y$ `Likes(x,y)` is *active voice*:
    Everyone likes everyone.
  - $\forall y \, \forall x$ `Likes(x,y)` is *passive voice*:
    Everyone is liked by everyone.

- Properties of quantifiers:
  - $\forall x \, \exists y$ is *not* the same as $\exists y \, \forall x$
  - $\exists x \, \forall y$ is *not* the same as $\forall y \, \exists x$

- Examples
  - $\forall x \, \exists y$ `Likes(x,y)` is *active voice*:
    Everyone likes someone.
  - $\exists y \, \forall x$ `Likes(x,y)` is *passive voice*:
    Someone is liked by everyone.

# FOL Syntax: Quantifiers

- Properties of quantifiers:
  - $\forall x \ P(x)$ is the same as $\neg\exists x \ \neg P(x)$
  - $\exists x \ P(x)$ is the same as $\neg\forall x \ \neg P(x)$

- General Identities
  - $\forall x \neg P \Leftrightarrow \neg\exists x P$
  - $\neg\forall x P \Leftrightarrow \exists x \neg P$
  - $\forall x P \Leftrightarrow \neg\exists x \neg P$
  - $\exists x P \Leftrightarrow \neg\forall x \neg P$
  - $\forall x P(x) \wedge Q(x) \Leftrightarrow \forall x P(x) \wedge \forall x Q(x)$
  - $\exists x P(x) \vee Q(x) \Leftrightarrow \exists x P(x) \vee \exists x Q(x)$

- Examples
  - $\forall x$ Likes(x,IceCream)

    Everyone likes ice cream.
  - $\neg\exists x \ \neg$Likes(x,IceCream)

    No one doesn't like ice cream.

    It's a double negative!

- Properties of quantifiers:
  - $\forall x \ P(x)$ when negated is $\exists x \ \neg P(x)$
  - $\exists x \ P(x)$ when negated is $\forall x \ \neg P(x)$

- Examples
  - $\forall x$ Likes(x,IceCream)

    Everyone likes ice cream.
  - $\exists x \ \neg$Likes(x,IceCream)

    Someone doesn't like ice cream.
  - This is from the application of de Morgan's law to the fully instantiated sentence

# FOL Syntax: Basics

- A **free variable** is a variable that isn't bound by a quantifier
  - `∃y Likes(x,y)`
    **x** is free, **y** is bound

- A **well-formed formula** is a sentence where all variables are quantified

# Summary so Far

- Constants: Bob, 2, Madison, …
- Functions: Income, Address, Sqrt, …
- Predicates: Sister, Teacher, <=, …
- Variables: x, y, a, b, c, …
- Connectives: $\wedge$ $\vee$ $\neg$ $\Rightarrow$ $\Leftrightarrow$
- Equality: $=$
- Quantifiers: $\forall$ $\exists$

- **Term**: Constant, variable, or Function($term_1$, ..., $term_n$)
  denotes an object in the world
  **Ground Term** has no variables
- **Atom**: Predicate($term_1$, ..., $term_n$), $term_1$ = $term_2$
  is smallest expression assigned a truth value
- **Sentence**: atom, quantified sentence with variables or complex sentence using connectives is assigned a truth value
- **Well-Formed Formula** (wff):
  sentence where all variables are quantified

# Fun with Sentences

Convert the following English sentences into FOL

- **Bob is a fish.**
  - What are the objects?
    Bob
  - What are the relations?
    is a fish

  **Answer:** `Fish(Bob)` a unary relation or property
- **Deb and Sue are women.** we'll be casual about plurals
- **Deb or Sue isn't a plant.** ambiguous?
- **Deb and Sue are friends.** use a function? predicate?

# Fun with Sentences

Convert the following English sentences into FOL

- **America bought Alaska from Russia.**
  - What are the objects?
    America, Alaska, Russia
  - What are the relations?
    bought(who, what, from) – an *n*-ary relation where *n* is 3

  **Answer:** `Bought(America, Alaska, Russia)`

- **Jim collects everything.**
  - What are the variables?
    everything $x$
  - How are they quantified?
    all, universal

  **Answer:** $\forall x$ `Collects(Jim,x)`

# Fun with Sentences

When to restrict the domain, e.g., to people:

- All: $\forall x\ (\mathrm{Person}(x) \wedge \ldots) \Rightarrow \ldots$
  - **things**: anything, everything, whatever
  - **people**: anybody, anyone, everybody, everyone, whoever
- Some (at least one): $\exists x\ \mathrm{Person}(x) \wedge \ldots \wedge \ldots$
  - **things**: something
  - **people**: somebody, someone
- None: $\neg\exists x\ \mathrm{Person}(x) \wedge \ldots \wedge \ldots$
  - **things**: nothing
  - **people**: nobody, no one

- **Somebody collects something.**
  - What are the variables?
    somebody **x** and something **y**
  - How are they quantified?
    at least one, existential

  **Answer:** $\exists x,y\ \mathrm{Person}(x) \wedge \mathrm{Collects}(x,y)$

Convert the following English sentences into FOL

- **Nothing collects anything.**
  - What are the variables and quantifiers?

    nothing **x** and anything **y**

    not one (i.e., not existential) and all (universal)

    **Answer:** $\neg\exists x\ \forall y\ \texttt{Collects}(x, y)$
  - Equivalent?

    Everything does not collect anything.

    **Answer:** $\forall x, y\ \neg\texttt{Collects}(x, y)$

- **All hoarders collect everything.**
  - How are ideas connected?

    being a hoarder implies collecting everything

  **Answer:** $\forall x, y\ \texttt{Horder}(x) \Rightarrow \texttt{Collects}(x, y)$

# Fun with Sentences

Convert the following English sentences into FOL

- **All stinky shoes are allowed.**
  - How are ideas connected?

    being a shoe and stinky implies it is allowed

  **Answer:** $\forall x\ (Shoe(x) \land Stinky(x)) \Rightarrow Allowed(x)$
- **No stinky shoes are allowed.** **Is this negative of above?**

  **Answer:** $\neg\exists x\ Shoe(x) \land Stinky(x) \land Allowed(x)$
  - Equivalent (carry negation through)?

    (All) Stinky shoes are not allowed.

    **Answer:** $\forall x\ (Shoe(x) \land Stinky(x)) \Rightarrow \neg Allowed(x)$

- **Any good amateur can beat some professional.**
  1. $\forall x$ [ ($x$ is a good amateur) $\Rightarrow$
     ($x$ can beat some professional) ]

OR

  2. (x can beat some professional) becomes
     $\exists y$ [ ($y$ is a professional) $\land$ ($x$ can beat $y$) ]

  **Answer:** $\forall x\ [(Amateur(x) \land GoodPlayer(x)) \Rightarrow \exists y\ (Professional(y) \land Beat(x,y))]$

# Inference Rules for FOL

- **Universal Elimination (UE)**
  variable substituted with ground term
  $\forall x\ Eats(Jim, x)$ infer $Eats(Jim, Cake)$

$$\frac{\forall v\ \alpha}{\text{SUBST}(\{v/g\}, \alpha)}$$

may be many ways to do this!

- **Existential Elimination (EE)**
  variable substituted with a new constant
  $\exists x\ Eats(Jim, x)$ infer $Eats(Jim, NewFood)$

$$\frac{\exists v\ \alpha}{\text{SUBST}(\{v/k\}, \alpha)}$$

*k* is a new term

- **These two inference rules enable the knowledge base to be propositionalized**
- **Then natural deduction can be done using inference rules for PL**

**Unification:**

A unifier of two atomic formulae is a substitution of terms **for variables** that makes them identical.

- Each variable has at most one associated term
- Substitutions are applied simultaneously

Unifier of $P(x, f(a), z)$ and $P(z, z, u)$ : $\{x/f(a), z/f(a), u/f(a)\}$

We can get the inference immediately if we can find a substitution $\alpha$ such that *King(x)* and *Greedy(x)* match *King(John)* and *Greedy(y)*

$\alpha$ = {x/John,y/John} works

# Example:

- **Anything anyone eats and not killed is food.**
- **Anil eats peanuts and still alive**
- **Harry eats everything that Anil eats.**
- **John likes all kind of food.**
- **Apple and vegetable are food**
- **Prove by resolution that:**
- **John likes peanuts.**

# Step-1: Conversion of Facts into FOL

We'll start by converting all of the given propositions to first-order logic.

a. $\forall x: food(x) \rightarrow likes(John, x)$

b. $food(Apple) \wedge food(vegetables)$

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

d. $eats\ (Anil, Peanuts) \wedge alive(Anil)$.

e. $\forall x : eats(Anil, x) \rightarrow eats(Harry, x)$

f. $\forall x: \neg killed(x) \rightarrow alive(x)$ **added predicates.**

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

h. $likes(John, Peanuts)$

## Step-2: Conversion of FOL into CNF

Converting FOL to CNF is essential in first-order logic resolution because CNF makes resolution proofs easier.

- **Eliminate all implication (→) and rewrite:**

1. $\forall x \neg food(x) \vee likes(John, x)$
2. $food(Apple) \wedge food(vegetables)$
3. $\forall x \forall y \neg [eats(x, y) \wedge \neg killed(x)] \vee food(y)$
4. $eats\ (Anil, Peanuts) \wedge alive(Anil)$
5. $\forall x \neg eats(Anil, x) \vee eats(Harry, x)$
6. $\forall x \neg [\neg killed(x)] \vee alive(x)$
7. $\forall x \neg alive(x) \vee \neg killed(x)$

- **Move negation (¬)inwards and rewrite**
    1. ∀x ¬ food(x) V likes(John, x)
    2. food(Apple) Λ food(vegetables)
    3. ∀x ∀y ¬ eats(x, y) V killed(x) V food(y)
    4. eats (Anil, Peanuts) Λ alive(Anil)
    5. ∀x ¬ eats(Anil, x) V eats(Harry, x)
    6. ∀x ¬killed(x) ] V alive(x)
    7. ∀x ¬ alive(x) V ¬ killed(x)
    8. likes(John, Peanuts).
- **Rename variables or standardize variables**
    1. ∀x ¬ food(x) V likes(John, x)
    2. food(Apple) Λ food(vegetables)
    3. ∀y ∀z ¬ eats(y, z) V killed(y) V food(z)
    4. eats (Anil, Peanuts) Λ alive(Anil)
    5. ∀x ¬ eats(Anil, x) V eats(Harry, x)
    6. ∀x ¬killed(x) ] V alive(x)
    7. ∀x ¬ alive(x) V ¬ killed(x)
    8. likes(John, Peanuts).
- **Eliminate existential instantiation quantifier by elimination.**
  We will eliminate existential quantifiers in this step, which is referred to as **Skolemization**. However, because there is no existential quantifier in this example problem, all of the assertions in this phase will be the same

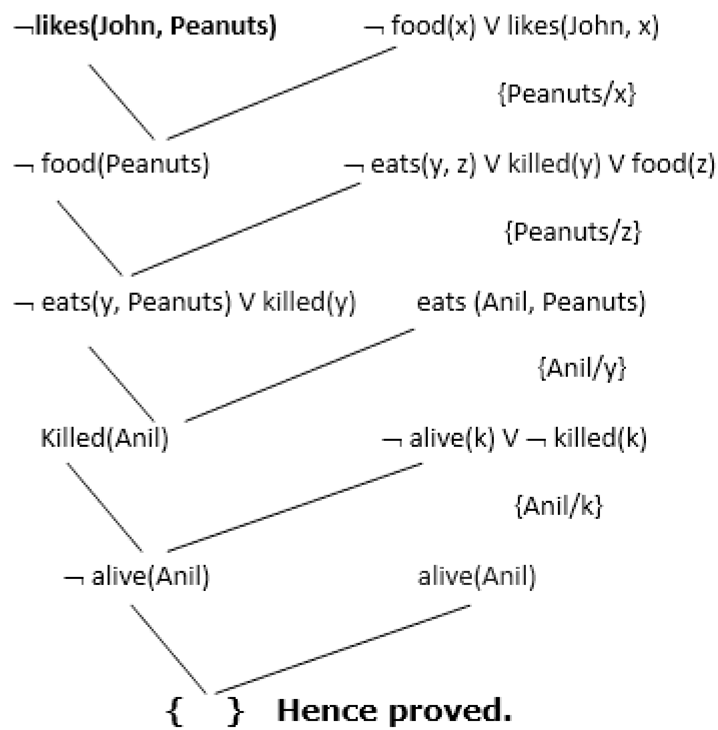
- **Drop Universal quantifiers.** We'll remove all universal quantifiers ∃ in this phase because none of the statements are implicitly quantified, therefore we don't need them

    1. $\neg$ food(x) V likes(John, x)
    2. food(Apple)
    3. food(vegetables)
    4. $\neg$ eats(y, z) V killed(y) V food(z)
    5. eats (Anil, Peanuts)
    6. alive(Anil)
    7. $\neg$ eats(Anil, w) V eats(Harry, w)
    8. killed(g) V alive(g)
    9. $\neg$ alive(k) V $\neg$ killed(k)
    10. likes(John, Peanuts).
        **[ Note: Statements "food(Apple) Λ food(vegetables)" and "eats (Anil, Peanuts) Λ alive(Anil)" can be written in two independent statements. ]**

- **Distribute conjunction ∧ over disjunction ¬.** This step will not make any change in this problem.

## Step 3: Reverse the statement that needs to be proven.

We will use negation to write the conclusion assertions in this statement, which will be written as "likes" (John, Peanuts)




As a result, the conclusion's negation has been demonstrated to constitute a total contradiction with the given collection of truths.

**Translate each of these statements into logical expressions using predicates, quantifiers, and logical connectives.**

- Let $P(x)$ be "$x$ is perfect"; let $F(x)$ be "$x$ is your friend"; and let the domain be all people.
- **a)** No one is perfect = everybody isn't perfect
- $\forall x \neg P(x)$
- **b)** Not (everyone is perfect).
- $\neg(\forall x\, P(x))$
- **c)** All your friends are perfect.
- $\forall x(F(x) \to P(x))$
- **d)** At least one of your friends is perfect.
- $\exists x(F(x) \wedge P(x))$
- **e)** Everyone is your friend and is perfect.
- $\forall x(F(x) \wedge P(x))$ or $(\forall x\, F(x)) \wedge (\forall x\, P(x))$
- **f )** everybody is not your friend or someone is not perfect. $(\forall x \neg F(x))) \vee (\exists x \neg P(x))$

## Towards Resolution for FOPL:

- Based on resolution for propositional logic
- Extended syntax: allow variables and quantifiers
- Define —clausal form" for first-order logic formulae (CNF)
- Eliminate quantifiers from clausal forms
- Adapt resolution procedure to cope with variables (unification)

## Conversion to CNF:

1. Eliminate implications and bi-implications as in propositional case
2. Move negations inward using De Morgan's laws

   plus rewriting $\neg \forall x P$ as $\exists x \neg P$ and $\neg \exists x P$ as $\forall x \neg P$
3. Eliminate double negations
4. Rename bound variables if necessary so each only occurs once

   e.g. $\forall x P(x) \vee \exists x Q(x)$ becomes $\forall x P(x) \vee \exists y Q(y)$
5. Use equivalences to move quantifiers to the left

   e.g. $\forall x P(x) \wedge Q$ becomes $\forall x\ (P(x) \wedge Q)$ where $x$ is not in $Q$

   e.g. $\forall x P(x) \wedge \exists y Q(y)$ becomes $\forall x \exists y (P(x) \wedge Q(y))$
6. Skolemise (replace each existentially quantified variable by a **new** term)

   $\exists x P(x)$ becomes $P(a_0)$ using a Skolem constant $a_0$ since $\exists x$ occurs at the outermost level

# Resolution in FOPL cntd..

$\forall x \exists y P(x, y)$ becomes $P(x, f_0(x))$ using a Skolem function $f_0$ since $\exists y$ occurs within $\forall x$

7. The formula now has only universal quantifiers and all are at the left of the formula: drop them
8. Use distribution laws to get CNF and then clausal form

**<u>Example:</u>**

1.) $\forall x\, [\forall y P(x, y) \rightarrow \neg \forall y (Q(x, y) \rightarrow R(x, y))]$

*<u>Solution:</u>*

1. $\forall x\, [\neg \forall y P(x, y) \vee \neg \forall y (\neg Q(x, y) \vee R(x, y))]$

2, 3. $\forall x\, [\exists y \neg P(x, y) \vee \exists y (Q(x, y) \wedge \neg R(x, y))]$

4. $\forall x\, [\exists y \neg P(x, y) \vee \exists z\, (Q(x, z) \wedge \neg R(x, z))]$

5. $\forall x \exists y \exists z\, [\neg P(x, y) \vee (Q(x, z) \wedge \neg R(x, z))]$

6. $\forall x\, [\neg P(x, f(x)) \vee (Q(x, g(x)) \wedge \neg R(x, g(x)))]$

7. $\neg P(x, f(x)) \vee (Q(x, g(x)) \wedge \neg R(x, g(x)))$

8. $(\neg P(x, f(x)) \vee Q(x, g(x))) \wedge (\neg P(x, f(x)) \vee \neg R(x, g(x)))$

8. $\{\neg P(x, f(x)) \vee Q(x, g(x)),\ \neg P(x, f(x)) \vee \neg R(x, g(x))\}$

# Resolution in FOPL cntd..

2.) $\neg \exists x \forall y \forall z\, ((P(y) \vee Q(z)) \rightarrow (P(x) \vee Q(x)))$

<u>*Solution:*</u>

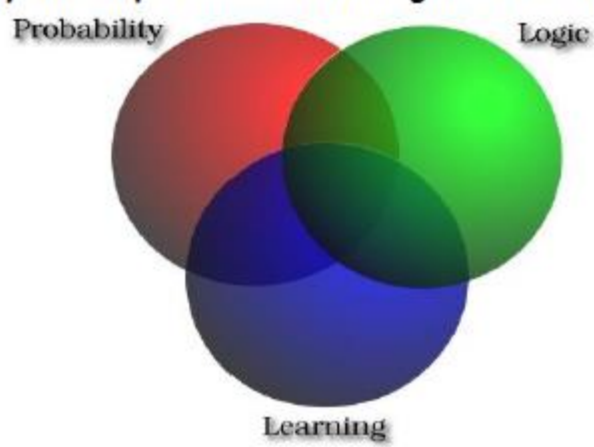
1. $\neg \exists x \forall y \forall z\, ( \neg (P(y) \vee Q(z)) \vee P(x) \vee Q(x))$
2. $\forall x \neg \forall y \forall z\, ( \neg (P(y) \vee Q(z)) \vee P(x) \vee Q(x))$
2. $\forall x \exists y \neg \forall z\, ( \neg (P(y) \vee Q(z)) \vee P(x) \vee Q(x))$
2. $\forall x \exists y \exists z \neg ( \neg (P(y) \vee Q(z)) \vee P(x) \vee Q(x))$
2. $\forall x \exists y \exists z\, ((P(y) \vee Q(z)) \wedge \neg (P(x) \vee Q(x)))$
6. $\forall x\, ((P(f(x)) \vee Q(g(x))) \wedge \neg P(x) \wedge \neg Q(x))$
7. $(P(f(x)) \vee Q(g(x)) \wedge \neg P(x) \wedge \neg Q(x)$
8. $\{P(f(x)) \vee Q(g(x)),\ \neg P(x),\ \neg Q(x)\}$

# Statistical Reasoning (The Bayesian Way)

## Probabilistic Logic Learning

### Probabilistic Approach to Uncertainty

- Logic agents almost never have access to the whole truth about their environment.
- There will always be questions to which a categorical answer (i.e., TRUE of FALSE) cannot be found.
- Will try to apply **probability theory** to deal with degree of belief about things.



## Bayes Theorem

- We have that $P(A,B) = P(A|B)\,P(B) = P(B|A)\,P(A)$ and therefore we can remove the joint probability to find that:

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

- Called **Bayes' Theorem** and provides a way to determine a conditional probability without the joint probability of A and B.

- Common to think of Bayes' Theorem in terms of updating our belief about a hypothesis A in the light of new evidence B.
  - Specifically, *our posterior belief P(A/B) is calculated by multiplying our prior belief P(A) by the likelihood P(B/A) that B will occur if A is true.*

- In many situations it is difficult to compute $P(A|B)$ directly, yet we might have information about $P(B|A)$.
  - Bayes' Theorem enables us to compute $P(A|B)$ in terms of $P(B|A)$.
  - This is what makes Bayes' Theorem so powerful.

# Chain Rule

- Recall that $P(A,B) = P(A|B)\,P(B)$.
- Can extend this formula to more variables.
- Example: For 3-variables:

  $$P(A,B,C) = P(A|\ B,C)\,P(B,C) = P(A|B,C)\,P(B|C)\,P(C)$$

- Example: For n-variables:

  $$P(A1,\ A2,\ \ldots,\ An) = P(A1|\ A2,\ \ldots,\ An)\,P(A2|\ A3,\ \ldots,\ An)\,P(An\text{-}1|An)\,P(An)$$

- In general we refer to this as the **chain rule**.
- This formula provides a means of calculating the full joint probability distribution.
- This formula is especially significant for **Bayesian Belief Nets** where/when many of the variables are conditionally independent (and the formula can be simplified.

# Independence and Conditional Independence

- Two events are independent if P(A ^ B) = P(A) P(B).
- If both events have a positive probability, then the statement of independence of events is equivalent to a statement of conditional independence:

  $P(A|B) = P(A)$ if and only if $P(B|A) = P(B)$ if and only if $P(A \wedge B) = P(A)\, P(B)$

- Can think of independence in the following way: knowledge that one event has occurred does not change the probability assigned to the other event.

# Independence


Age and Gender are independent.

$$P(A,G) = P(G)P(A)$$

$$P(A|G) = P(A) \qquad A \perp G$$
$$P(G|A) = P(G) \qquad G \perp A$$

$$P(A,G) = P(G|A)\ P(A) = P(G)P(A)$$
$$P(A,G) = P(A|G)\ P(G) = P(A)P(G)$$

# Bayesian networks

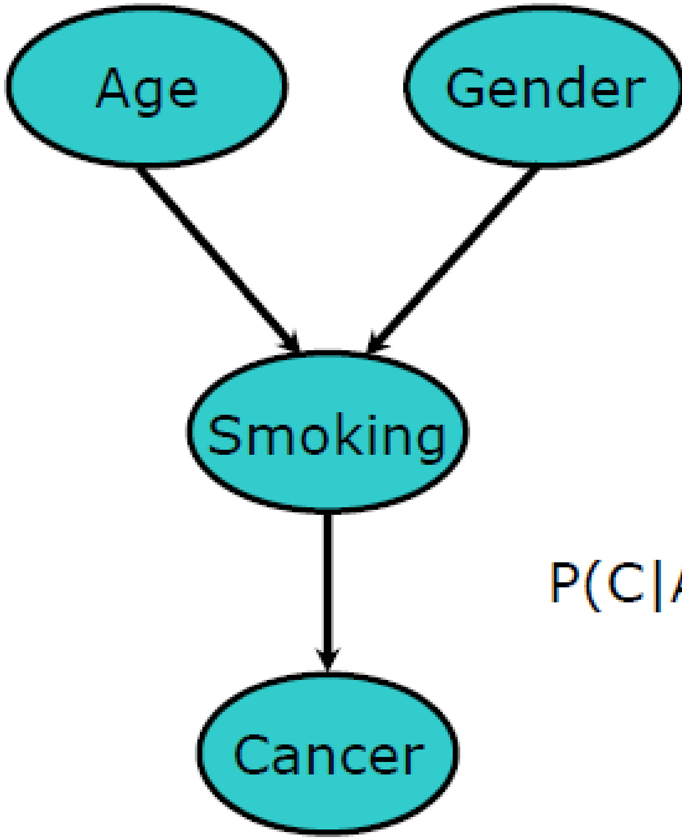
- A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

- Syntax:
  - a set of nodes, one per variable
  - a directed, acyclic graph (link ≈ "directly influences")
  - a conditional distribution for each node given its parents:

$$\mathbf{P}(X_i \mid Parents(X_i))$$

- In the simplest case, conditional distribution represented as a conditional probability table (CPT) giving the distribution over $X_i$ for each combination of parent values

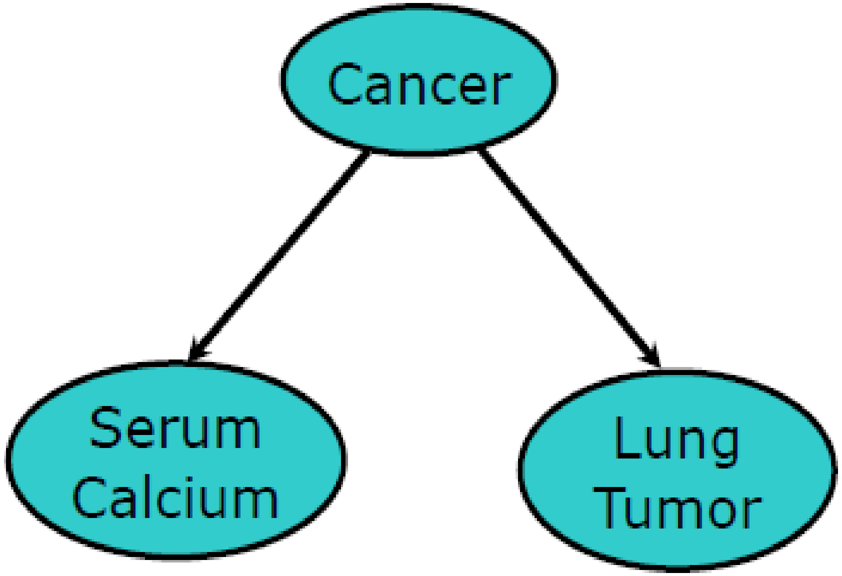
# Conditional Independence




Cancer is independent of Age and Gender given Smoking.

$P(C|A,G,S) = P(C|S) \quad C \perp A,G \mid S$

# More Conditional Independence: Naïve Bayes



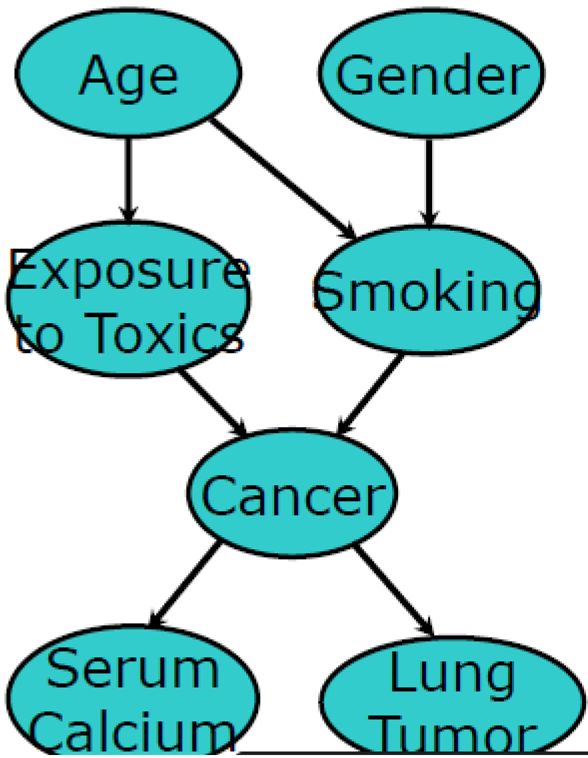

Serum Calcium and Lung Tumor are dependent

Serum Calcium is independent of Lung Tumor, given Cancer

$$P(L|SC,C) = P(L|C)$$

# Bayesian Network(Directed Acyclic Graph →DAG)

Put it all together




$$P(A,G,E,S,C,L,SC) =$$

$$P(A) \cdot P(G) \cdot$$

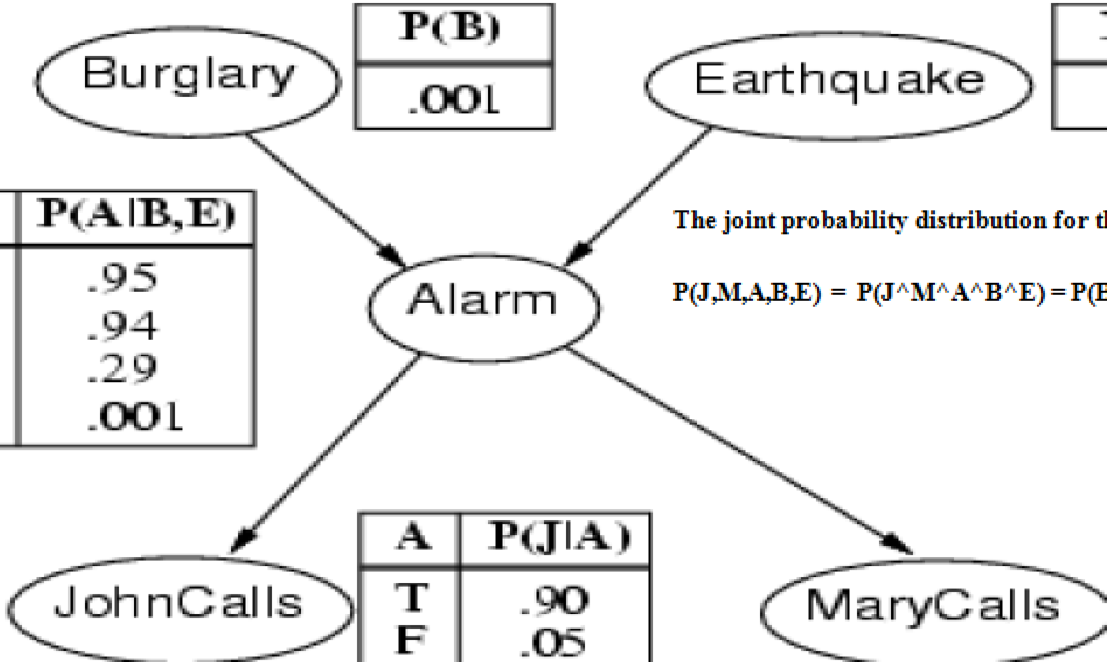
$$P(E \mid A) \cdot P(S \mid A,G) \cdot$$

$$P(C \mid E,S) \cdot$$

$$P(SC \mid C) \cdot P(L \mid C)$$

# Example

- I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?
- Variables: *Burglary, Earthquake, Alarm, JohnCalls, MaryCalls*
- Network topology reflects "causal" knowledge:
  - A burglar can set the alarm off
  - An earthquake can set the alarm off
  - The alarm can cause Mary to call
  - The alarm can cause John to call




| P(B) |
|---|
| .001 |

| P(E) |
|---|
| .002 |

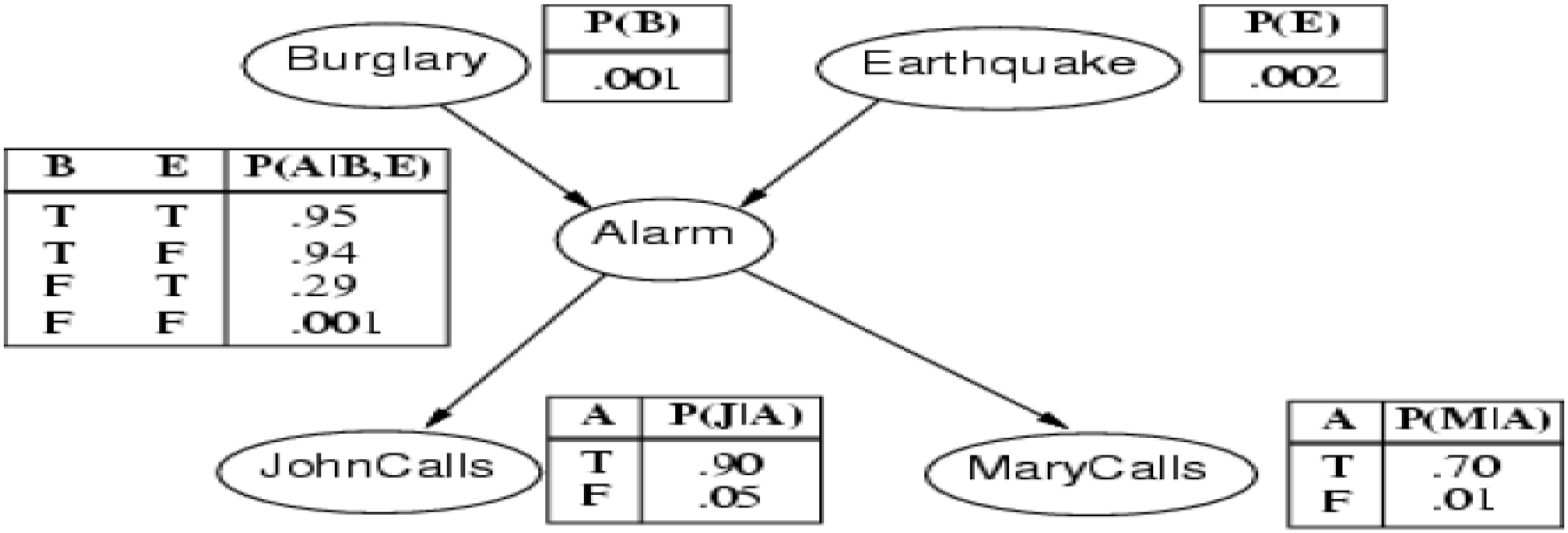
| B | E | P(A\|B,E) |
|---|---|---|
| T | T | .95 |
| T | F | .94 |
| F | T | .29 |
| F | F | .001 |

| A | P(J\|A) |
|---|---|
| T | .90 |
| F | .05 |

| A | P(M\|A) |
|---|---|
| T | .70 |
| F | .01 |

**The joint probability distribution for the above network is given by:**

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




| P(B) |
|---|
| .001 |

| P(E) |
|---|
| .002 |

| B | E | P(A\|B,E) |
|---|---|---|
| T | T | .95 |
| T | F | .94 |
| F | T | .29 |
| F | F | .001 |

| A | P(J\|A) |
|---|---|
| T | .90 |
| F | .05 |

| A | P(M\|A) |
|---|---|
| T | .70 |
| F | .01 |

**The joint probability distribution for the above network is given by:**

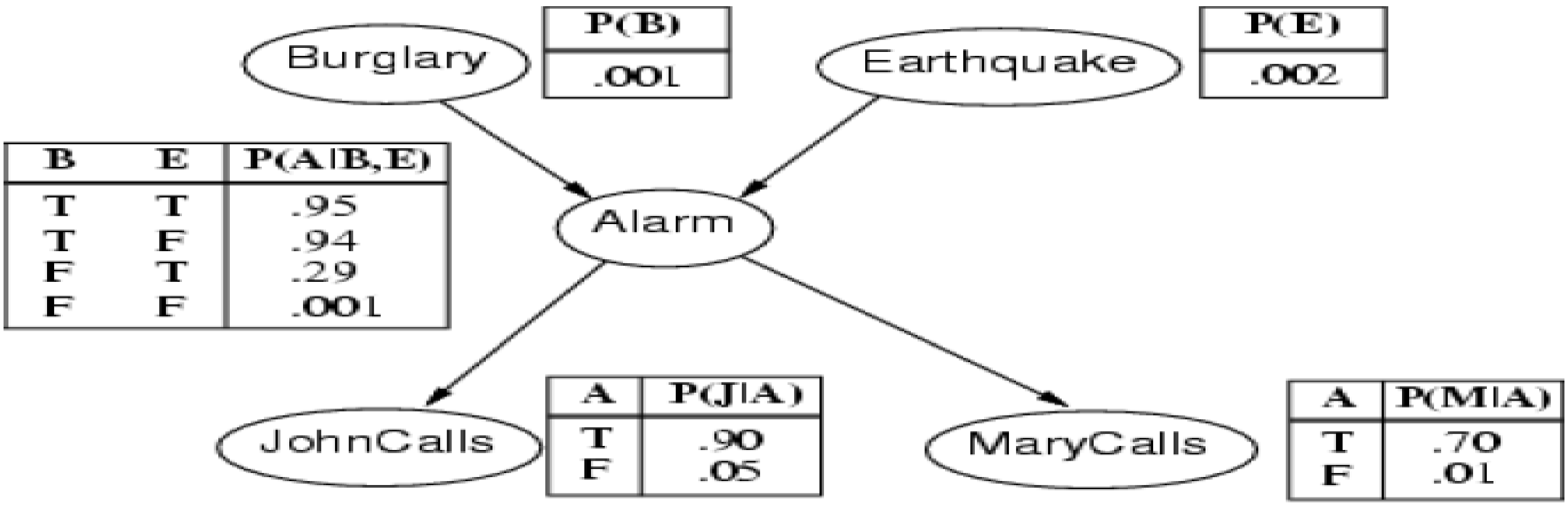
**P(J,M,A,B,E) = P(J^M^A^B^E) = P(B) * P(E) * P(A | B , E) * P(J|A) * P(M|A)**

What is the probability that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both John and Merry call?

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j \mid a)\, P(m \mid a)\, P(a \mid \neg b, \neg e)\, P(\neg b)\, P(\neg e)$$

$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$$

$$= 0.00062$$

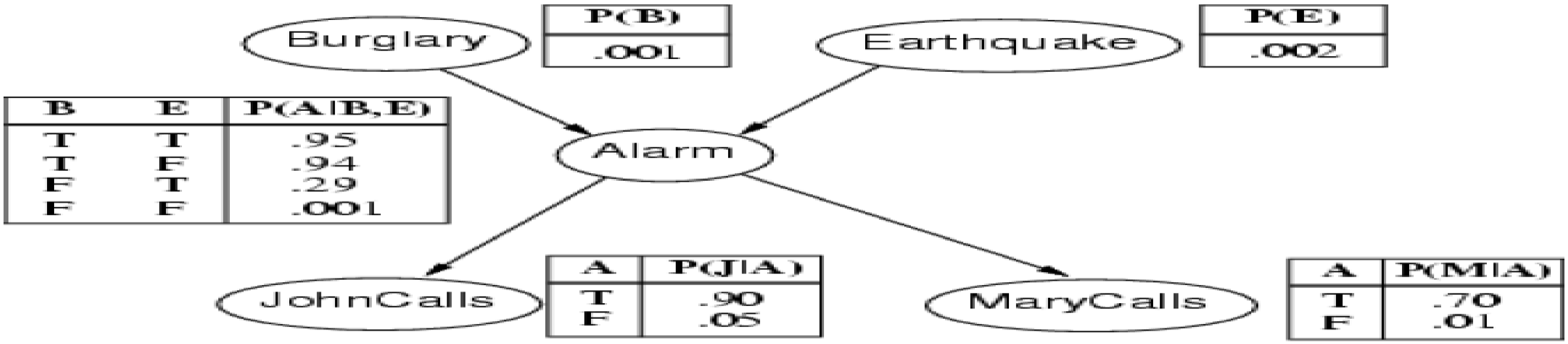



What is the probability of john calling mary ?

$$P(j) = P(j \mid a) * P(a) + P(j \mid \neg a) * P(\neg a)$$

$$= P(j \mid a) * [\, P(a \mid b, e) * P(b, e) + P(a \mid \neg b, e) * P(\neg b, e) + P(a \mid b, \neg e) * P(b, \neg e) + P(a \mid \neg b, \neg e) * P(\neg b, \neg e)\,] + P(j \mid \neg a) * [\, P(\neg a \mid b, e) * P(b, e) + P(\neg a \mid \neg b, e) * P(\neg b, e) + P(\neg a \mid b, \neg e) * P(b, \neg e) + P(\neg a \mid \neg b, \neg e) * P(\neg b, \neg e)\,]$$

$$= 0.90 * 0.00252 + 0.005 * 0.9974 = 0.0521$$




What is the probability of john calling mary ?

$P(j) = P(j \mid a) * P(a) + P(j \mid \neg a) * P(\neg a)$

**$= P(j \mid a) * [\, P(a \mid b, e) * P(b, e) + P(a \mid \neg b, e) * P(\neg b, e) + P(a \mid b, \neg e) * P(b, \neg e) + P(a \mid \neg b, \neg e) * P(\neg b, \neg e) \,] + P(j \mid \neg a) * [\, P(\neg a \mid b, e) * P(b, e) + P(\neg a \mid \neg b, e) * P(\neg b, e) + P(\neg a \mid b, \neg e) * P(b, \neg e) + P(\neg a \mid \neg b, \neg e) * P(\neg b, \neg e) \,]$**

Note → Where $P(b, e) = P(b) * P(e)$ since B and E are independent events from the graph networks

$P(a \mid b, e) = 0.95$

$P(\neg a \mid b, e) = 1 - P(a \mid b, e) = 1 - 0.95 = 0.05$

$P(\neg a \mid b, \neg e) = 1 - P(a \mid b, \neg e) = 1 - 0.94 = 0.06$

# References


- Rich and knight
- Insights on AI
- AIIMA
- Chuck S Dyer slides
- Utoronto slides
- https://www.javatpoint.com/ai-techniques-of-knowledge-representation
- Probabilistic Graphical Models: **Daphne** Koller and Nir Friedman
- https://tutorialforbeginner.com/ai-resolution-in-first-order-logic
- https://shishirkant.com/resolution-method-in-ai/