

## Unit III

### CONJUNCTIVE NORMAL FORM (CNF) or CLAUSAL NORMAL FORM

- A formula is in conjunctive normal form (CNF) or clausal normal form if it is a conjunction of one or more clauses, where a clause is a disjunction of literals; otherwise put, it is an AND of ORs.
- Every propositional formula can be converted into an equivalent formula that is in CNF. This transformation is based on rules about logical equivalences: the double negative law, De Morgan's laws, and the distributive law.
- As a normal form, it is useful in automated theorem proving
- Conjunctive Normal Form (CNF): A WFF is in CNF format when it is a *conjunction of disjunctions of literals*.
- Example:  $(\neg P \vee Q \vee R) \wedge (S \vee P \vee T \vee \neg R) \wedge (Q \vee S)$
- A clause is the disjunction of many things. The units that make up a clause are called literals.

#### Steps to convert PL into CNF

1. Remove  $\leftrightarrow$  using rule:
  - a)  $\alpha \leftrightarrow \beta \equiv (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$
2. Remove  $\rightarrow$  using rule:
  - a)  $\alpha \rightarrow \beta \equiv \neg \alpha \vee \beta$
3. Move  $\neg$  (negation) inwards
  - a)  $\neg(\alpha \vee \beta) \equiv \neg \alpha \wedge \neg \beta$
  - b)  $\neg(\alpha \wedge \beta) \equiv \neg \alpha \vee \neg \beta$
  - c)  $\neg(\neg \alpha) \equiv \alpha$
4. Apply distributive and/ or commutative law
  - a)  $\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$
  - b)  $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$
  - c)  $\alpha \wedge \beta \equiv \beta \wedge \alpha$
  - d)  $\alpha \vee \beta \equiv \beta \vee \alpha$

#### EXAMPLE:

$$P \leftrightarrow (Q \vee R)$$

1. Eliminate  $\leftrightarrow$ , replacing  $\alpha \leftrightarrow \beta$  with  $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$ .  
 $(P \rightarrow (Q \vee R)) \wedge ((Q \vee R) \rightarrow P)$
2. Eliminate  $\rightarrow$ , replacing  $\alpha \rightarrow \beta$  with  $\neg \alpha \vee \beta$ .  
 $(\neg P \vee Q \vee R) \wedge (\neg(Q \vee R) \vee P)$
3. Move  $\neg$  inwards using de Morgan's rules and double-negation:  
 $(\neg P \vee Q \vee R) \wedge ((\neg Q \wedge \neg R) \vee P)$
4. Apply distributive law ( $\wedge$  over  $\vee$ ) and flatten:  
 $(\neg P \vee Q \vee R) \wedge (\neg Q \vee P) \wedge (\neg R \vee P)$

### RESOLUTION

In propositional logic, the procedure for producing a proof by resolution of proposition P with respect to a set of axioms F is called **RESOLUTION**.

Resolution for CNF – applied to a special type of wffs: *conjunction of clauses*.

*Literal* – either an atom (e.g., P) or its negation ( $\neg P$ ).

*Clause* – disjunction of literals (e.g.,  $(P \vee Q \vee \neg R)$ ).

## ALGORITHM: PROPOSITIONAL RESOLUTION

1. Convert all the propositions of F to clause form
2. Negate P and convert the result to clause form. Add it to the set of clauses obtained in step 1.
3. Repeat until either a contradiction is found or no progress can be made:
  - (a) Select two clauses. Call these the parent clauses.
  - (b) Resolve them together. The resulting clause, called the resolvent, will be the disjunction of all of the literals of both of the parent clauses with the following exception: If there are any pairs of literals L and  $\neg L$  such that one of the parent clauses contains L and the other contains  $\neg L$ , then select one such pair and eliminate both L and  $\neg L$  from the resolvent.
  - (c) If the resolvent is the empty clause, then a contradiction has been found. If it is not then add it to the set of clauses available to the procedure.

### Limitations of Propositional Logic

Consider the following arguments:

- All dogs are faithful.
- Tommy is a dog.
- Therefore, Tommy is faithful.

How to represent and infer this in Propositional Logic?

Suppose, P considers “all dogs are faithful”.

Q considers “Tommy is a dog”.

But,  $P \wedge Q \rightarrow$  “Tommy is faithful”?

No, we cannot infer this in Propositional Logic.

Another example,

Tom is a hardworking student.

Hardworking (Tom)

Tom is an intelligent student

Intelligent (Tom)

If Tom is a hardworking and Tom is intelligent then Tom scores high marks.

In PL,

$\text{Hardworking (Tom)} \wedge \text{Intelligent (Tom)} \rightarrow \text{score\_high\_marks (Tom)}.$

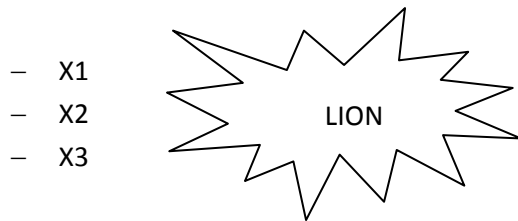
But, what about John and Jill? If we could write instead “All students who are hardworking and intelligent scores high marks.

Unfortunately, the given statement cannot be written in Propositional Logic because PL structure don't allow this.

### FIRST ORDER LOGIC or PREDICATE LOGIC

- FOL is a generalization of PL that allows us to express and infer arguments in infinite model like
  - All men are mortal
  - Some birds cannot fly
  - At least one planet has life on it
- Each sentence, or statement, is broken down into a **subject** and a **predicate**.
- In each sentence, we actually talk about something. So in the sentence “Pinky is a cat” where “Pinky” is a **subject**. And we are giving property to the subject which is called **predicate**. So in the same sentence “cat” is the **predicate**. The predicate modifies or defines the properties of the subject. In first-order logic, a predicate can only refer to a single subject.

- A sentence in first-order logic is written in the form  $P(x)$ , where  $P$  is the predicate and  $x$  is the subject, represented as a variable. As in the sentence “Pinky is a cat”, we can write it in FOL as cat (pinky) which actually we called methods. In more generalized form, cat (x) means x is a cat.
- Sometimes, subject is not a single element. It becomes a group of objects. Example: “Every Lion drinks coffee”. So we do have a Universe of Discourse “Lion”, so whenever we refer variable “x” then that means “x” is one of the Lion which represents the connection of “x” and UoD.

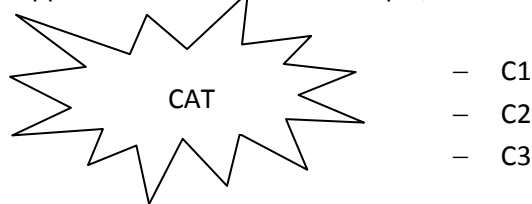


Where each  $x_1, x_2, x_3$  are particular element of LION set, then “every Lion drinks coffee” is same as:

drinks ( $X_1$ , coffee)  $\wedge$  drinks ( $X_2$ , coffee)  $\wedge$  drinks ( $X_3$ , coffee)

Now if we do have infinite lions then we cannot mention each element separately. How many subjects would be in UoD?

Again, suppose consider another example, “Some cats are intelligent”



Statement either refers  $C_1, C_2$  or  $C_3$  is intelligent.

Intelligent ( $C_1$ )  $\vee$  Intelligent ( $C_2$ )  $\vee$  Intelligent ( $C_3$ ).

If there is no cat who is intelligent, then the whole statement would be FALSE. Again if UoD of cat is infinite so again it would be imposing to ORing the statements.

- In first-order logic, a sentence can be structured using the universal quantifier (symbolized  $\forall$ ) or the existential quantifier ( $\exists$ ).

So for the above example,

$\forall x \{ \text{drinks } (x, \text{coffee}) \}$

$\exists x \{ \text{intelligent } (x) \}$

- Every complete “sentence” contains two parts: a “subject” and a “predicate”.
  - The subject is what (or whom) the sentence is about.
  - The predicate tell something about the subject.

Example: the sky is blue, predicate is “is blue” and subject is “sky” and can be represented as  $\text{Blue}(\text{sky})$

### **Syntax of FOL:**

1. Connectives
2. Quantifiers
3. Constants
4. Variables
5. Functions
6. Predicates

User defines these primitives:

**Constant** symbols: representing individuals in the world. E.g., Mary, 3

**Function** symbols map individuals to individuals. E.g., father-of(Mary) = John, color-of(Sky) = Blue

**Predicate** symbols map from individuals to truth values. E.g., greater(5,3), green(Grass), color(Grass, Green)

FOL supplies these primitives:

**Variable** symbols. E.g., x, y

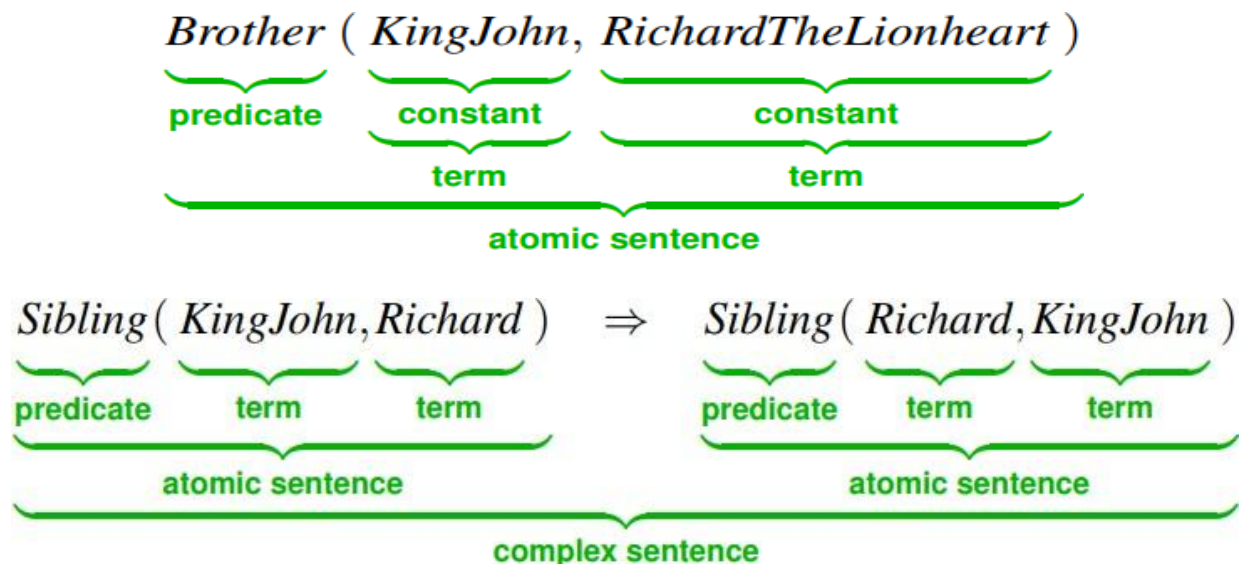
**Connectives**

- Same as in propositional logic: not ( $\neg$ ), and ( $\wedge$ ), or ( $\vee$ ), implies ( $\rightarrow$ ), iff ( $\leftrightarrow$ )

**Quantifiers:** Universal ( $\forall$ ) and Existential ( $\exists$ )

Sentences are built up from terms and atoms:

- A term (denoting a real-world individual) is a constant symbol, a variable symbol, or an n-place function of n terms. For example, x and  $f(x_1, \dots, x_n)$  are terms, where each  $x_i$  is a term.
- An atom (which has value true or false) is either an n-place predicate of n terms, or, if P and Q are atoms, then  $\neg P$ ,  $P \vee Q$ ,  $P \wedge Q$ ,  $P \rightarrow Q$ ,  $P \leftrightarrow Q$  are atoms
- A sentence is an atom, or, if P is a sentence and x is a variable, then  $(\forall x)P$  and  $(\exists x)P$  are sentences
- A well-formed formula (wff) is a sentence containing no "free" variables, i.e., all variables are "bound" by universal or existential quantifiers. E.g.,  $(\forall x) P(x,y)$  has x bound as a universally quantified variable, but y is free.



## INFERENCE RULES FOR FOL

Inference rules for PL apply to FOL as well.

**New (sound) inference rules for use with quantifiers:**

### **Universal Elimination**

If  $(\forall x)P(x)$  is true, then  $P(c)$  is true, where c is a constant in the domain of x. For example, from  $(\forall x)\text{eats}(\text{Ziggy}, x)$  we can infer  $\text{eats}(\text{Ziggy}, \text{IceCream})$ . The variable symbol can be replaced by any ground term, i.e., any constant symbol or function symbol applied to ground terms only.

## Existential Introduction

If  $P(c)$  is true, then  $(\exists x)P(x)$  is inferred. For example, from  $\text{eats}(\text{Ziggy}, \text{IceCream})$  we can infer  $(\exists x)\text{eats}(\text{Ziggy}, x)$ . All instances of the given constant symbol are replaced by the new variable symbol. Note that the variable symbol cannot already exist anywhere in the expression.

## Existential Elimination

From  $(\exists x)P(x)$  infer  $P(c)$ . For example, from  $(\exists x)\text{eats}(\text{Ziggy}, x)$  infer  $\text{eats}(\text{Ziggy}, \text{Cheese})$ . Note that the variable is replaced by a brand new constant that does not occur in this or any other sentence in the Knowledge Base. In other words, we don't want to accidentally draw other inferences about it by introducing the constant. All we know is there must be some constant that makes this true, so we can introduce a brand new one to stand in for that (unknown) constant.

## Entailment

- Entailment means that one thing follows from another:  $KB \models \alpha$
- Knowledge base  $KB$  entails sentence  $\alpha$  if and only if  $\alpha$  is true in all worlds where  $KB$  is true E.g., the  $KB$  containing "the Giants won" and "the Reds won" entails "Either the Giants won or the Reds won" E.g.,  $x+y = 4$  entails  $4 = x+y$
- Entailment is a relationship between sentences (i.e., syntax) that is based on semantics

## Proof System

A proof is a sequence of sentences, where each sentence is either a premise or a sentence derived from earlier sentences in the proof by one of the rules of inference.

The last sentence is the theorem (also called goal or query) that we want to prove.

## Unification

if we can find a substitution  $\theta$  such that  $\text{King}(x)$  and  $\text{Greedy}(x)$  match  $\text{King}(\text{John})$  and  $\text{Greedy}(y)$

p	q	$\theta$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(\text{John}, \text{Jane})$	$\{x/\text{Jane}\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(y, \text{OJ})$	$\{x/\text{OJ}, y/\text{John}\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(y, \text{Mother}(y))$	$\{x/\text{Mother}(\text{John}), y/\text{John}\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(x, \text{OJ})$	$\{\text{fail}\}$

Convert the following sentences into FOL:

**Every gardener likes the sun.**

$$\forall x \text{gardener}(x) \rightarrow \text{likes}(x, \text{Sun})$$

**You can fool some of the people all of the time.**

$$\exists x \forall t \text{person}(x) \wedge \text{time}(t) \rightarrow \text{can-fool}(x, t)$$

**You can fool all of the people some of the time.** (two ways)

$$\forall x \exists t (\text{person}(x) \rightarrow \text{time}(t) \wedge \text{can-fool}(x, t))$$

$$\forall x (\text{person}(x) \rightarrow \exists t (\text{time}(t) \wedge \text{can-fool}(x, t)))$$

**All purple mushrooms are poisonous.**

$\forall x (\text{mushroom}(x) \wedge \text{purple}(x)) \rightarrow \text{poisonous}(x)$

**No purple mushroom is poisonous.** (two ways)

$\neg \exists x \text{purple}(x) \wedge \text{mushroom}(x) \wedge \text{poisonous}(x)$

$\forall x (\text{mushroom}(x) \wedge \text{purple}(x)) \rightarrow \neg \text{poisonous}(x)$

**There are exactly two purple mushrooms.**

$\exists x \exists y \text{mushroom}(x) \wedge \text{purple}(x) \wedge \text{mushroom}(y) \wedge \text{purple}(y) \wedge \neg(x=y) \wedge \forall z (\text{mushroom}(z) \wedge \text{purple}(z)) \rightarrow ((x=z) \vee (y=z))$

**Bush is not tall.**

$\neg \text{tall}(\text{Bush})$

**X is above Y iff X is on directly on top of Y or there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.**

$\forall x \forall y \text{above}(x,y) \leftrightarrow (\text{on}(x,y) \vee \exists z (\text{on}(x,z) \wedge \text{above}(z,y)))$

## CONJUNCTIVE NORMAL FORMULA

FOL TO CNF: First Order Logic: Conversion to CNF

1. Eliminate bi-conditionals and implications:
  - Eliminate  $\leftrightarrow$ , replacing  $\alpha \leftrightarrow \beta$  with  $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$ .
  - Eliminate  $\Rightarrow$ , replacing  $\alpha \Rightarrow \beta$  with  $\neg \alpha \vee \beta$ .
2. Move  $\neg$  inwards:
  - $\neg(\forall x p) \equiv \exists x \neg p$ ,
  - $\neg(\exists x p) \equiv \forall x \neg p$ ,
  - $\neg(\alpha \vee \beta) \equiv \neg \alpha \wedge \neg \beta$ ,
  - $\neg(\alpha \wedge \beta) \equiv \neg \alpha \vee \neg \beta$ ,
  - $\neg \neg \alpha \equiv \alpha$ .
3. Standardize variables apart by renaming them: each quantifier should use a different variable.
4. Skolemize: each existential variable is replaced by a Skolem constant or Skolem function of the enclosing universally quantified variables.
  - For instance,  $\exists x \text{Rich}(x)$  becomes  $\text{Rich}(G1)$  where  $G1$  is a new Skolem constant.
  - “Everyone has a heart”  $\forall x \text{Person}(x) \Rightarrow \exists y \text{Heart}(y) \wedge \text{Has}(x, y)$  becomes  $\forall x \text{Person}(x) \Rightarrow \text{Heart}(H(x)) \wedge \text{Has}(x, H(x))$ , where  $H$  is a new symbol (Skolem function).
5. Drop universal quantifiers
  - For instance,  $\forall x \text{Person}(x)$  becomes  $\text{Person}(x)$ .
6. Distribute  $\wedge$  over  $\vee$ :
  - $(\alpha \wedge \beta) \vee \gamma \equiv (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$ .

Exercise:

1. Convert “Everybody who loves all animals is loved by someone” to CNF

$(\forall x)[(\forall y)\{\text{Animal}(y) \rightarrow \text{Loves}(x,y)\} \rightarrow (\exists y) \text{Loves}(y,x)]$

Solution:

1. Eliminate implications:  $\forall x [\neg \forall y \neg \text{Animal}(y) \vee \text{Loves}(x, y)] \vee [\exists y \text{Loves}(y, x)]$
2. Move  $\neg$  inwards
  - a)  $\forall x [\exists y \neg(\neg \text{Animal}(y) \vee \text{Loves}(x, y))] \vee [\exists y \text{Loves}(y, x)]$
  - b)  $\forall x [\exists y \neg \neg \text{Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists y \text{Loves}(y, x)]$  (De Morgan)
  - c)  $\forall x [\exists y \text{Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists y \text{Loves}(y, x)]$  (double negation)

3. Standardize variables:  $\forall x [\exists y \text{ Animal}(y) \wedge \neg \text{Loves}(x, y)] \vee [\exists z \text{ Loves}(z, x)]$
4. Skolemization:  $\forall x [\text{Animal}(F(x)) \wedge \neg \text{Loves}(x, F(x))] \vee [\text{Loves}(G(x), x)]$
5. Drop universal quantifiers:  $[\text{Animal}(F(x)) \wedge \neg \text{Loves}(x, F(x))] \vee [\text{Loves}(G(x), x)]$
6. Distribute  $\vee$  over  $\wedge$ :  $[\text{Animal}(F(x)) \vee \text{Loves}(G(x), x)] \wedge [\neg \text{Loves}(x, F(x)) \vee \text{Loves}(G(x), x)]$

### RESOLUTION IN PREDICATE LOGIC

Two literals are contradictory if one can be unified with the negation of the other. For example  $\text{man}(x)$  and  $\text{man}(\text{Himalayas})$  are contradictory since  $\text{man}(x)$  and  $\neg \text{man}(x)$  can be unified. In predicate logic unification algorithm is used to locate pairs of literals that cancel out. It is important that if two instances of the same variable occur, then they must be given identical substitutions. The resolution algorithm for predicate logic as follows

Let  $F$  be a set of given statements and  $S$  is a statement to be proved.

1. Covert all the statements of  $F$  to clause form.
2. Negate  $S$  and convert the result to clause form. Add it to the set of clauses obtained in 1.
3. Repeat until either a contradiction is found or no progress can be made or a predetermined amount of effort has been expended.
  - a. Select two clauses. Call them parent clauses.
  - b. Resolve them together. The resolvent will be the disjunction of all of these literals of both clauses. If there is a pair of literals  $T_1$  and  $T_2$  such that one parent clause contains  $T_1$  and the other contains  $\neg T_2$  and if  $T_1$  and  $T_2$  are unifiable, then neither  $T_1$  nor  $T_2$  should appear in the resolvent. Here  $T_1$  and  $T_2$  are called complimentary literals.
  - c. If the resolvent is the empty clause, then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

Resolution Example: Anyone passing his history exams and winning the lottery is happy. But anyone who studies or is lucky can pass all his exams. John did not study but John is lucky. Anyone who is lucky wins the lottery. Is John happy?

#### **Example 1:**

Convert the following sentence into predicate logic and then prove "Is someone smiling?" using resolution:

1. All people who are graduating are happy
2. All happy people smile
3. Someone is graduating

FOL:

- $\forall x \{ \text{graduating}(x) \rightarrow \text{happy}(x) \}$
- $\forall x \{ \text{happy}(x) \rightarrow \text{smile}(x) \}$
- $\exists x \{ \text{graduate}(x) \}$
- Prove:  $\exists x \{ \text{smile}(x) \}$

Negate the conclusion:  $\neg \exists w \{ \text{smile}(w) \}$

FOL to CNF:

Step 1:

- $\forall x \{ \neg \text{graduating}(x) \vee \text{happy}(x) \}$
- $\forall x \{ \neg \text{happy}(x) \vee \text{smile}(x) \}$
- $\exists x \{ \text{graduate}(x) \}$
- $\neg \exists w \{ \text{smile}(w) \}$



Step 2: standardize variable

- $\forall x\{rgraduating(x) \vee happy(x)\}$
- $\forall y\{rhappy(y) \vee smile(y)\}$
- $\exists z\{graduate(z)\}$
- $\forall w(rsmile(w))$

Step 3: skolemization

- $\forall x\{rgraduating(x) \vee happy(x)\}$
- $\forall y\{rhappy(y) \vee smile(y)\}$
- $graduate(A)$
- $\forall w(rsmile(w))$

Step 4: drop universal

- $rgraduating(x) \vee happy(x)$
- $rhappy(y) \vee smile(y)$
- $graduate(A)$
- $rsmile(w)$

Resolution tree

- If facts 'F' is to be proved then it start with "rF"
- It contradicts all the other rule in KB
- It process stops when it returns NULL clause.

## Example 2:

Convert the following sentence into predicate logic and then prove "Was Marcus loyal to Caesar?" using resolution:

1. Marcus was a man.
2. Marcus was a Pompeian.
3. All Pompeians were Romans.
4. Caesar was a ruler.
5. All Romans were either loyal to Caesar or hated him.
6. Everyone is loyal to someone.
7. People only try to assassinate rulers they are not loyal to.
8. Marcus tried to assassinate Caesar.

### Solution:

The facts described by these sentences can be represented as a set of wff's in predicate logic as follows:

1. Marcus was a man.  
 $\text{man}(\text{Marcus})$
2. Marcus was a Pompeian.  
 $\text{Pompeian}(\text{Marcus})$
3. All Pompeians were Romans.  
 $\forall x: \text{Pompeian}(x) \rightarrow \text{Roman}(x)$
4. Caesar was a ruler.  
 $\text{ruler}(\text{Caesar})$
5. All Romans were either loyal to Caesar or hated him.  
 $\forall x: \text{Roman}(x) \rightarrow \text{loyalto}(x, \text{Caesar}) \vee \text{hate}(x, \text{Caesar})$
6. Everyone is loyal to someone.  
 $\forall x: \rightarrow y: \text{loyalto}(x, y)$
7. People only try to assassinate rulers they are not loyal to.  
 $\forall x: \forall y: \text{person}(x) \wedge \text{ruler}(y) \wedge \text{tryassassinate}(x, y) \rightarrow \neg \text{loyalto}(x, y)$
8. Marcus tried to assassinate Caesar.  
 $\text{tryassassinate}(\text{Marcus}, \text{Caesar})$

Additional:  $\forall x: \text{man}(x) \rightarrow \text{person}(x)$

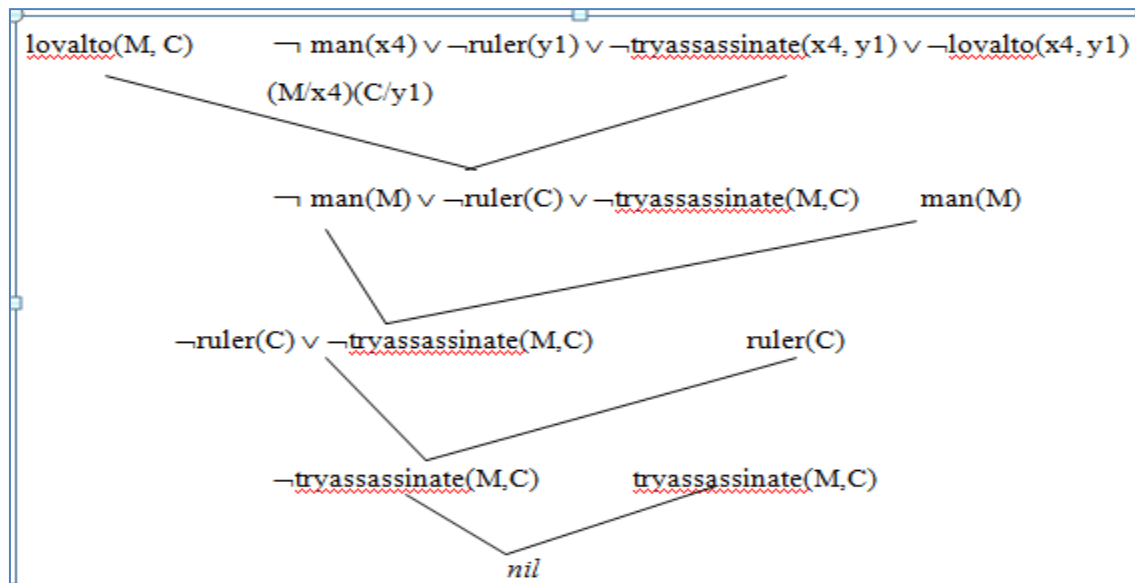
$\forall x: \text{rman}(x) \vee \text{person}(x)$

9.  $\text{rman}(x) \vee \text{person}(x)$

CNF:

- o  $\text{man}(\text{Marcus})$
- o  $\text{Pompeian}(\text{Marcus})$
- o  $\neg \text{Pompeian}(x1) \vee \text{Roman}(x1)$
- o  $\text{ruler}(\text{Caesar})$
- o  $\neg \text{Roman}(x2) \vee \text{loyalto}(x2, \text{Caesar}) \vee \text{hate}(x2, \text{Caesar})$
- o  $\text{loyalto}(x3, S1(x3))$
- o  $\neg \text{person}(x4) \vee \neg \text{ruler}(y1) \vee \neg \text{tryassassinate}(x4, y1) \vee \neg \text{loyalto}(x4, y1)$
- o  $\text{tryassassinate}(\text{Marcus}, \text{Caesar})$

Resolution tree:



# 1. Convert the following sentences to FOPL

- Jack owns a dog
- Every dog owner is an animal lover.
- No animal lover kills an animal.
- Either Jack or Curiosity killed the cat, who is named Tuna

Also prove by resolution - Did curiosity kill the cat

- $(\exists x) \text{Dog}(x) \wedge \text{Owns}(\text{Jack}, x)$
- $(\forall x) ((\exists y) \text{Dog}(y) \wedge \text{Owns}(x, y)) \rightarrow \text{AnimalLover}(x)$
- $(\forall x) \text{AnimalLover}(x) \rightarrow ((\forall y) \text{Animal}(y) \rightarrow \neg \text{Kills}(x, y))$
- $\text{Kills}(\text{Jack}, \text{Tuna}) \vee \text{Kills}(\text{Curiosity}, \text{Tuna})$
- $\text{Cat}(\text{Tuna})$
- $(\forall x) \text{Cat}(x) \rightarrow \text{Animal}(x)$
- $\text{Kills}(\text{Curiosity}, \text{Tuna}) \leftarrow \text{GOAL}$

# 2. Translate the following sentences into first order logic.

- All dogs are mammals
- Fido is a dog
- Fido is a mammal
- All mammals produce milk

1a) The translation into first order logic is fairly straightforward:

- $\forall X (\text{dog}(X) \rightarrow \text{mammal}(X)).$
- $\text{dog}(\text{fido}).$
- $\text{mammal}(\text{fido}).$
- $\forall X (\text{mammal}(X) \rightarrow \text{milk}(X)).$

# 3. Using propositional linear resolution, show the following propositional sentence is unsatisfiable. Convert this sentence to clause form and derive the empty clause using resolution-

$$(P \vee Q \vee \neg R) \vee ((\neg R \vee Q \vee P) \Rightarrow (R \vee Q) \wedge \neg Q \wedge \neg P)$$

# 4. Represent the following sentence in the Predicate form "All the children like sweets".

# 5. Represent the following sentences in first-order logic, using a consistent vocabulary (which you must define):

- Not all students take both History and Biology'
- Only one student failed History'
- Only one student failed both History and Biology'
- The best score in History was better than the best score in Biology.
- Every person who dislikes all vegetarians is smart
- No person likes a smart vegetarian'
- There is a woman who likes all men who are not vegetarians.
- There is a barber who shaves all men in town who do not shave themselves.
- No person likes a professor unless the professor is smart.
- politicians can fool some of the people all of the time, and they can fool all of the people some of the time, but they can't fool all of the people all of the time'

Give a predicate calculus sentence such that every world in which it is true contains exactly one object.

# 6. Represent the sentence "All Germans speak the same languages" in predicate calculus. Use $\text{speaks}(x, l)$ , meaning that person 'x' speaks language 'l'.

7.

Consider the following sentences :

- John likes all kinds of food.
  - Apples are food.
  - Chicken is food
  - Anything anyone eats and isn't killed alive.
  - Sue eats everything bill eats.
- (i) Translate these sentences into formulas in predicate logic.
- (ii) Prove that John likes peanuts using backward chaining.

**Solution:**

**FOL:**

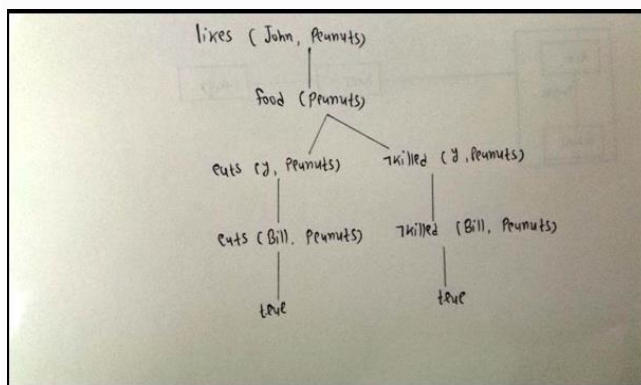
- a)  $\forall x \text{ food}(x) \rightarrow \text{likes}(\text{John}, x)$
- b)  $\text{food}(\text{Apple})$
- c)  $\text{food}(\text{Chicken})$
- d)  $\forall x \exists y [ \text{eats}(y, x) \wedge \neg \text{killed}(y, x) \rightarrow \text{food}(x) ]$
- e)  $\text{eats}(\text{Bill}, \text{peanuts}) \wedge \text{alive}(\text{Bill}, \text{peanuts})$
- f)  $\forall x \text{ eats}(\text{Bill}, x) \Rightarrow \text{eats}(\text{Sue}, x)$

**Clausal Form:**

- a.  $\neg \text{food}(x) \vee \text{likes}(\text{John}, x)$
- b.  $\text{food}(\text{Apple})$
- c.  $\text{food}(\text{Chicken})$
- d.  $\neg \text{eats}(y, x) \vee \text{killed}(y, x) \vee \text{food}(x)$
- e.  $\text{eats}(\text{Bill}, \text{peanuts})$
- f.  $\neg \text{killed}(\text{Bill}, \text{peanuts})$
- g.  $\neg \text{eats}(\text{Bill}, x) \vee \text{eats}(\text{Sue}, x)$

Resolve with Sub-goal	
	$\neg \text{likes}(\text{John}, \text{peanuts})$
1	$\neg \text{food}(\text{peanuts})$
4	$\neg \text{eats}(y, \text{peanuts}) \vee \text{killed}(y, \text{peanuts})$
5b	$\neg \text{eats}(y, \text{peanuts})$
5a	Null clauses

**Using backward chaining:**



8.

Convert the following English sentences into predicate logic :

- (i) Every fisherman likes the river.
- (ii) No red flower is poisonous.
- (iii) You can fool some of the people all of the time.
- (iv) There are exactly two purple mushrooms.
- (v) X is above Y if X is on directly on top of Y or else there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.

OR

Convert the following to predicate calculus

- A) Every fisherman likes the river.
- B) No red flower is poisonous.
- C) You can fool some of the people all of the time.
- D) All red flowers are poisonous.
- E) You can fool all of the people some of the time.
- F) There are exactly two purple mushrooms.
- G) Shree is not tall.
- H) X is above Y if X is on directly on top of Y or else there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.

Solution:

- A)  $\forall (x) \text{fisherman}(x) \Rightarrow \text{likes}(x, \text{river})$
- B)  $\neg (\exists x) \text{red}(x) \wedge \text{flower}(x) \wedge \text{poisonous}(x)$  or, equivalently,  
 $(\forall x) (\text{flower}(x) \wedge \text{red}(x)) \Rightarrow \neg \text{poisonous}(x)$
- C)  $(\exists x) (\text{person}(x) \wedge (\forall t) (\text{time}(t) \Rightarrow \text{can-fool}(x, t)))$
- D)  $(\forall x) (\text{flower}(x) \wedge \text{red}(x)) \Rightarrow \text{poisonous}(x)$
- E)  $(\forall x) (\text{person}(x) \Rightarrow (\exists t) (\text{time}(t) \wedge \text{can-fool}(x, t)))$
- F)  $(\exists x) (\exists y) \text{mushroom}(x) \wedge \text{purple}(x) \wedge \text{mushroom}(y) \wedge \text{purple}(y) \wedge$   
 $\neg (x=y) \wedge (\forall z) (\text{mushroom}(z) \wedge \text{purple}(z)) \Rightarrow ((x=z) \vee (y=z))$
- G)  $\neg \text{tall}(\text{Shree})$
- H)  $(\forall x) (\forall y) \text{above}(x, y) (\text{on}(x, y) \vee (\exists z) (\text{on}(x, z) \wedge \text{above}(z, y)))$

9. Consider the facts:

- a) Anyone whom Mary loves is a football star.
- b) Any student who does not pass does not play.
- c) John is a student.
- d) Any student who does not study does not pass.
- e) Anyone who does not play is not a football star.

Prove using resolution "If John does not study, then, Mary does not love John".

### Predicate logic:

- a)  $\forall x (\text{LOVES}(\text{Mary}, x) \rightarrow \text{STAR}(x))$
- b)  $\forall x (\text{STUDENT}(x) \wedge \neg \text{PASS}(x) \rightarrow \neg \text{PLAY}(x))$
- c)  $\text{STUDENT}(\text{John})$
- d)  $\forall x (\text{STUDENT}(x) \wedge \neg \text{STUDY}(x) \rightarrow \neg \text{PASS}(x))$
- e)  $\forall x (\neg \text{PLAY}(x) \rightarrow \neg \text{STAR}(x))$

### Clause form:

- a)  $\neg \text{LOVES}(\text{Mary}, x) \vee \text{STAR}(x)$
- b)  $\neg \text{STUDENT}(x) \vee \text{PASS}(x) \vee \neg \text{PLAY}(x)$
- c)  $\text{STUDENT}(\text{John})$
- d)  $\neg \text{STUDENT}(x) \vee \text{STUDY}(x) \vee \neg \text{PASS}(x)$
- e)  $\text{PLAY}(x) \vee \neg \text{STAR}(x)$

Resolve with Sub-goal	
	$\neg \text{STUDY}(\text{John}) \vee \text{LOVES}(\text{Mary}, \text{John})$
1	$\neg \text{STUDY}(\text{John}) \vee \text{STAR}(\text{John})$
5	$\neg \text{STUDY}(\text{John}) \vee \text{PLAY}(\text{John})$
4	$\text{PLAY}(\text{John}) \vee \neg \text{PASS}(\text{John}) \vee \neg \text{STUDENT}(\text{John})$
3	$\text{PLAY}(\text{John}) \vee \neg \text{PASS}(\text{John})$
2	$\neg \text{STUDENT}(\text{John})$
3	Null clause

## INFERENCE RULES

### Deductive inference rule:

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

A

A  $\rightarrow$  B

B

-----

Example:

It is raining.

If it is raining, the street is wet.

The street is wet.

-----

Forward chaining starts with the available data and uses inference rules to extract more data until the goal is reached. Also called data-driven knowledge extraction.

### Abductive inference rule:

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

B

A  $\rightarrow$  B

A

-----

Example:

The street is wet.

If it is raining, the street is wet.

It is raining.

Backward chaining is done in the backward direction. The system selects a goal state and rules whose then portion has the goal state as conclusion. It establishes sub-goals to be satisfied for the goal state to be true. Also called goal-driven knowledge extraction.

## PROBABILITY THEORY

Conditional probability,  $P(A|B)$ , indicates the probability of event A given that we know event B has occurred.

Syntax and semantics of graphical models for representing probability distributions:

- Bayesian networks (built on top of directed graphs)
- Markov networks (built on top of undirected graphs)

### Independent events

The difference between the two examples is that in the first one, the two events are independent while in the second they are not.

- Events A and B are independent if
$$P(A \cap B) = P(A) \cdot P(B).$$

### Conditional probability

Let A and B be events, with  $P(B) > 0$ .

- The conditional probability  $P(A|B)$  of A given B is given by
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- Notice: If A and B are independent, then

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

### Probabilistic Reasoning based on Bayes' Theorem:

- In probability theory, Bayes' theorem (alternatively Bayes' law or Bayes' rule) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.
- Bayes' theorem is named after Reverend Thomas Bayes.
- Bayes' theorem is stated mathematically as the following equation:

THE PROBABILITY OF "B" BEING TRUE GIVEN THAT "A" IS TRUE

THE PROBABILITY OF "A" BEING TRUE

THE PROBABILITY OF "A" BEING TRUE GIVEN THAT "B" IS TRUE

THE PROBABILITY OF "B" BEING TRUE

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

### EXAMPLE:

1. At a certain university, 4% of men are over 6 feet tall and 1% of women are over 6 feet tall. The total student population is divided in the ratio 3:2 in favour of women. If a student is selected at random from among all those over six feet tall, what is the probability that the student is a woman?



**Solution**

Let  $M = \{\text{Student is Male}\}$ ,  $F = \{\text{Student is Female}\}$ , (note that  $M$  and  $F$  partition the sample space of students),  $T = \{\text{Student is over 6 feet tall}\}$ . We know that  $P(M) = 2/5$ ,  $P(F) = 3/5$ ,  $P(T|M) = 4/100$  and  $P(T|F) = 1/100$ . We require  $P(F|T)$ . Using Bayes' Theorem we have:

$$\begin{aligned} P(F|T) &= \frac{P(T|F)P(F)}{P(T|F)P(F) + P(T|M)P(M)} \\ &= \frac{\frac{1}{100} \times \frac{3}{5}}{\frac{1}{100} \times \frac{3}{5} + \frac{4}{100} \times \frac{2}{5}} \\ &= \frac{3}{11} \end{aligned}$$

2. A factory production line is manufacturing bolts using three machines, A, B and C. Of the total output, machine A is responsible for 25%, machine B for 35% and machine C for the rest. It is known from previous experience with the machines that 5% of the output from machine A is defective, 4% from machine B and 2% from machine C. A bolt is chosen at random from the production line and found to be defective. What is the probability that it came from (a) machine A (b) machine B (c) machine C?

**Solution**

Let  $D = \{\text{bolt is defective}\}$ ,  $A = \{\text{bolt is from machine A}\}$ ,  $B = \{\text{bolt is from machine B}\}$ ,  $C = \{\text{bolt is from machine C}\}$ . We know that  $P(A) = 0.25$ ,  $P(B) = 0.35$  and  $P(C) = 0.4$ . Also  $P(D|A) = 0.05$ ,  $P(D|B) = 0.04$ ,  $P(D|C) = 0.02$ . A statement of Bayes' Theorem for three events  $A, B$  and  $C$  is

$$\begin{aligned} P(A|D) &= \frac{P(D|A)P(A)}{P(D|A)P(A) + P(D|B)P(B) + P(D|C)P(C)} \\ &= \frac{0.05 \times 0.25}{0.05 \times 0.25 + 0.04 \times 0.35 + 0.02 \times 0.4} \\ &= 0.362 \end{aligned}$$

Similarly

$$\begin{aligned} P(B|D) &= \frac{0.04 \times 0.35}{0.05 \times 0.25 + 0.04 \times 0.35 + 0.02 \times 0.4} \\ &= 0.406 \end{aligned}$$

$$\begin{aligned} P(C|D) &= \frac{0.02 \times 0.4}{0.05 \times 0.25 + 0.04 \times 0.35 + 0.02 \times 0.4} \\ &= 0.232 \end{aligned}$$

3. Machines A and B produce 10% and 90% respectively of the production of a component intended for the motor industry. From experience, it is known that the probability that machine A produces a defective component is 0.01 while the probability that machine B produces a defective component is 0.05. If a component is selected at random from a day's production and is found to be defective, find the probability that it was made by (a) machine A; (b) machine B.



Let  $A = \{\text{item from machine } A\}$ ,  $B = \{\text{item from machine } B\}$ ,  $D = \{\text{item is defective}\}$ . We know that:  $P(A) = 0.1$ ,  $P(B) = 0.9$ ,  $P(D|A) = 0.01$ ,  $P(D|B) = 0.05$ .

(a)

$$\begin{aligned} P(A|D) &= \frac{P(D|A)P(A)}{P(D|A)P(A) + P(D|B)P(B)} \\ &= \frac{0.01 \times 0.1}{0.01 \times 0.1 + 0.05 \times 0.9} \\ &= 0.02 \end{aligned}$$

(b) Similarly  $P(B|D) = 0.98$

### UTILITY THEORY

- The main idea of Utility Theory is: an agent's preferences over possible outcomes can be captured by a function that maps these outcomes to a real number; the higher the number the more that agent likes that outcome. The function is called a utility function.
- Utility Theory uses the notion of Expected Utility (EU) as a value that represents the average utility of all possible outcomes of a state, weighted by the probability that the outcome occurs.
- The agent can use probability theory to reason about uncertainty. The agent can use utility theory for rational selection of actions based on preferences. Decision theory is a general theory for combining probability with rational decisions

**Decision theory = Probability theory + Utility theory**

- The other key concept of Utility Theory is known as the Principle of Maximum Utility (MEU) which states that a rational agent should choose an action that maximizes the agent's expected utility

### HIDDEN MARKOV MODEL (HMM)

- In probability theory, a Markov model is a stochastic model used to model randomly changing systems. It is assumed that future states depend only on the current state, not on the events that occurred before it.
- A Markov chain is a sequence of states such that the  $(n+1)^{\text{th}}$  state is independent of all previous states if the  $n^{\text{th}}$  state is known. That is, you can predict the distribution of the next state if I tell you the current state, without telling how I got to the current state.
- A hidden Markov model is a Markov chain for which the state is only partially observable. In other words, observations are related to the state of the system, but they are typically insufficient to precisely determine the state.

**Note:** The main weakness of Markov networks is their inability to represent induced and non-transitive dependencies; two independent variables will be directly connected by an edge, merely because some other variable depends on both. As a result, many useful independencies go unrepresented in the network. To overcome this deficiency, Bayesian networks use the richer language of directed graphs, where the directions of the arrows permit us to distinguish genuine dependencies from spurious dependencies induced by hypothetical observations.

## BAYESIAN NETWORK

A probability theory is called a Bayesian network when the underlying graph is directed and a Markov network/Markov random field when the underlying graph is undirected.

A Bayesian network is a DAG:

- Each node corresponds to a random variables
- Directed edges link nodes
- Edge goes from parent to child
- Each node has a conditional probability dist.
- $P(X_i | \text{Parents}(X_i))$
- Topology of network represents causality

Example: The Alarm Problem by Pearl 1990

You have a new burglar alarm installed. It is reliable about detecting burglary, but responds to minor earthquakes. Two neighbors (John, Mary) promise to call you at work when they hear the alarm. John always calls when hears alarm, but confuses alarm with phone ringing (and calls then also). Mary likes loud music and sometimes misses alarm! Given evidence about who has and hasn't called, estimate the probability of a burglary.

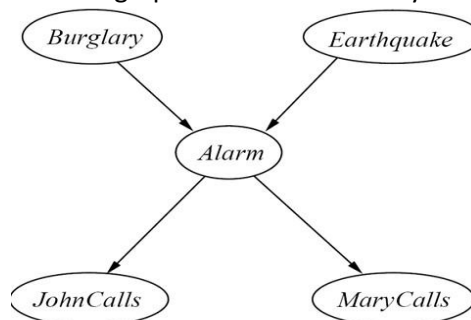
Represent problem using 5 binary variables:

- B = a burglary occurs at your house
- E = an earthquake occurs at your house
- A = the alarm goes off
- J = John calls to report the alarm
- M = Mary calls to report the alarm

Bayesian networks (or belief networks) are a graphical and compact way to represent uncertain knowledge, based on this idea.

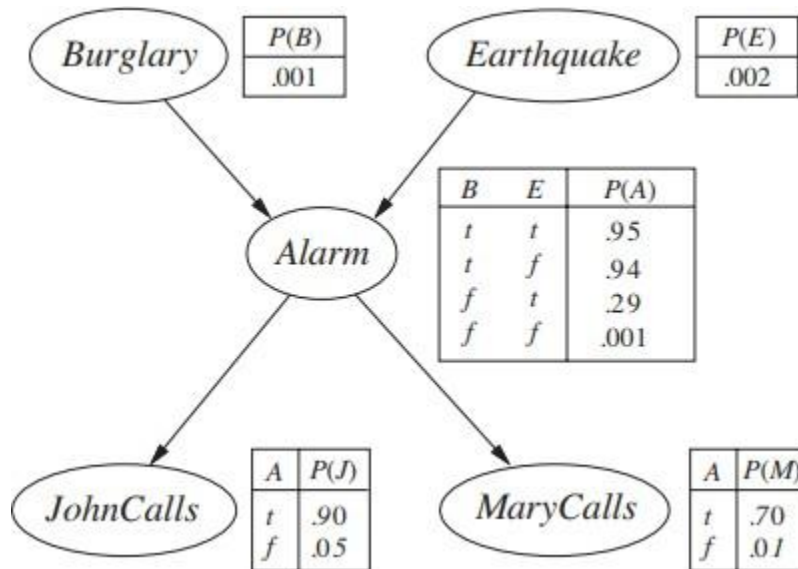
### Constructing the Bayesian Network

1. Order the variables in terms of causality (may be a partial order)  
e.g.,  $\{E, B\} \rightarrow \{A\} \rightarrow \{J, M\}$
2. Use these assumptions to create the graph structure of the Bayesian network



3. Fill in Conditional Probability Table (CPT)
  - a. One for each node
  - b.  $2^p$  entries, where p is the number of parents

**Note:** Where do these probabilities come from: expert knowledge, data (retrieval frequency estimates)



A Bayesian network is directed acyclic graph (DAG) with the following components:

1. A set of nodes, one for each random variable of the “world” represented
2. A set of directed arcs connecting nodes
  - a. If there is an arc connecting X with Y , we say that X is a parent of Y (parents(X) denotes the set of parents variable of X)
  - b. From the concept of parent variable, we define also the concepts of ancestors and descendants of a random variable.
3. Each node  $X_i$  has an associated conditional probability distribution (CPT)  $P(X_i | \text{parents}(X_i))$ 
  - a. If  $X_i$  is Boolean, we usually omit the probability of the false value.

General form of Bayesian Network:

$$P(X_1, X_2, \dots, X_N) = \prod_i P(X_i | \text{parents}(X_i))$$

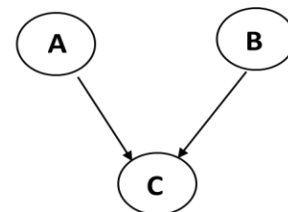
↖
↗

The full joint distribution                      The graph-structured approximation

$$P(X_1, X_2, \dots, X_N) = \prod_i P(X_i | \text{parents}(X_i))$$

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



An example: the probability that the alarm has gone off, both John and Mary call the police, but nothing has happened is:

$$P(J, M, A, \neg B, \neg E) = P(J|A)P(M|A)P(A|\neg B, \neg E)P(\neg B)P(\neg E) = 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 = 0.00062$$

For example, the probability of a Burglary, knowing that both John and Mary called the police i.e., compute  $P(\text{burglary} | \text{john, mary})$

It allows you to say things like: if you know the state of the sprinkler (on or off) and the state of the rain (rainy or not), then knowing whether it's cloudy doesn't affect the probability that the grass is wet.

The main weakness of Markov networks is their inability to represent induced and non-transitive dependencies; two independent variables will be directly connected by an edge, merely because some other variable depends on both. As a result, many useful independencies go unrepresented in the network. To overcome this deficiency, Bayesian networks use the richer language of directed graphs, where the directions of the arrows permit us to distinguish genuine dependencies from spurious dependencies induced by hypothetical observations.