**Knowledge Base (KB):** This is where agent stores its knowledge about world. Represented in various ways = logical statements, rules, or probabilistic models. It contains facts, rules, constraints, and beliefs about environment.

**Inference System:** Allows agent to reason with knowledge in KB to conclude. It includes mechanisms for querying KB, making inferences, & make appropriate actions based on current state of world & agent's goals.

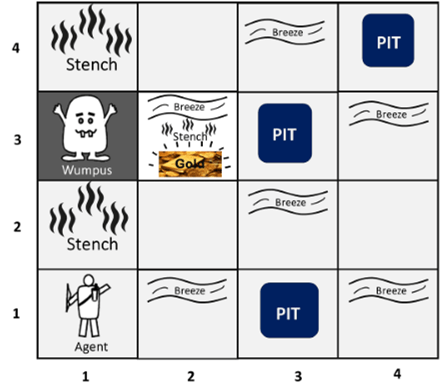
**Capabilities of a knowledge-based agent:-**

* *Representing states, actions, etc.*
* *Incorporating new percepts:* It should add it to what it already knows.
* *Updating the internal representation of the world:* keep its understanding up-to-date.
* *Deducing hidden properties of the world:* Figure out things based on what it knows.
* *“ ” appropriate actions:* Decide how to achieve its goals or react to what's happening.

**Logics** are formal languages for representing info such that conclusions can be drawn.

Syntax: defines all possible sequences of symbols to constitute sentences of the language.

Semantics: determines facts to which the sentences refer. It defines "meaning" of sentences.

**Entailment** is the generation or discovery that a new sentence is TRUE given existing sentences. Relationship between sentences (i.e., syntax) that is based on semantics. 

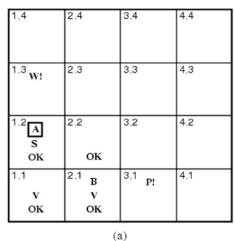
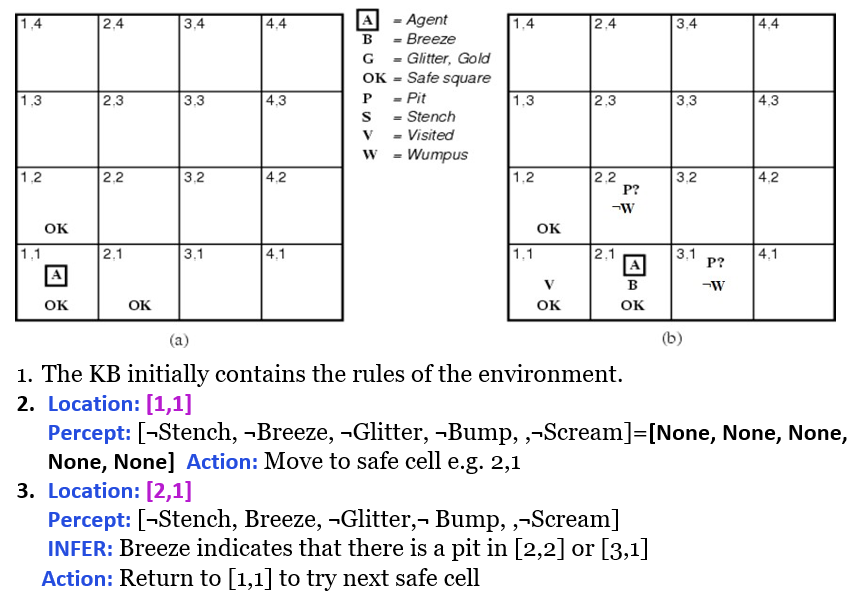
**The Wumpus world** is a scenario used to demonstrate the capabilities of a KB agent and knowledge representation in AI. It consists of a cave with rooms, some containing a deadly Wumpus, bottomless pits, and possibly gold.

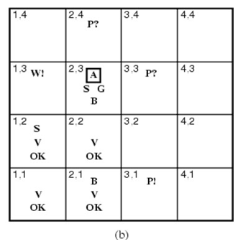
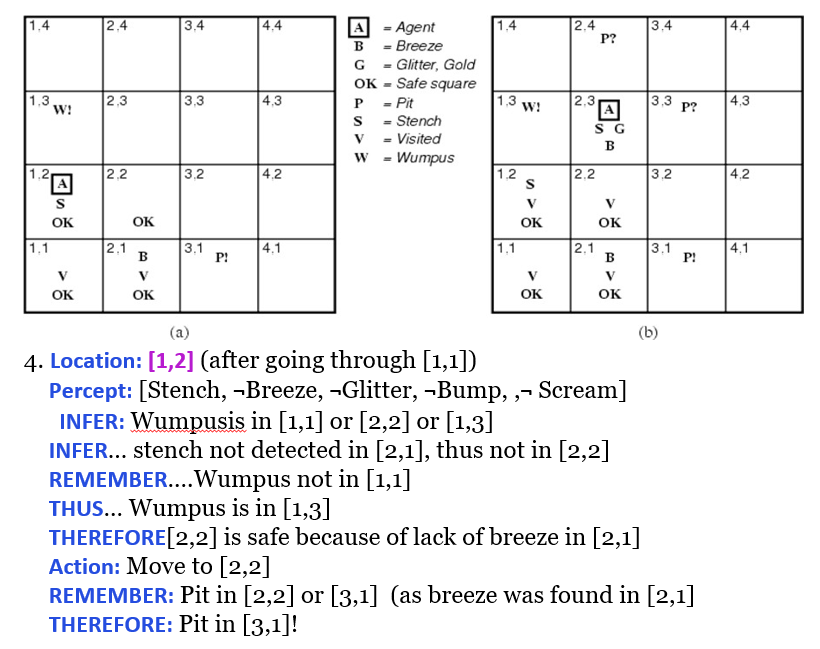
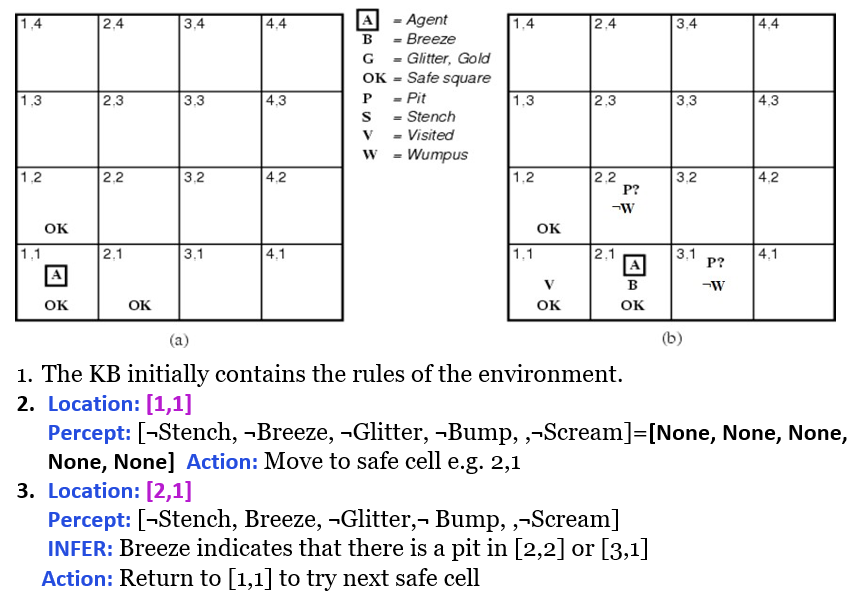
*Goal:* find the gold, exit the cave w/o being eaten by the Wumpus or falling into pits. It uses percepts from its environment, like stench (indicating the Wumpus), breeze (indicating pits), & glitter (indicating gold), to make decisions.

*PEAS Description:*

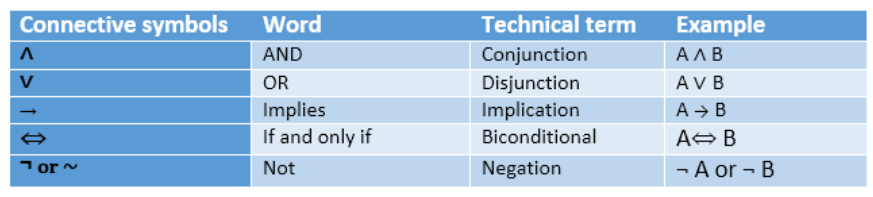
* **P:** +1000 for escaping with gold, -1000 for being eaten or falling, -1 for each action, -10 for shooting.
* **E:** 4x4 grid of rooms. Squares adjacent to Wumpus = smelly, Squares adjacent to pit = breezy, Glitter = gold is in same square, Gold picked up by reflex, can’t be dropped. Shooting kills Wumpus if facing it, it screams. Shooting uses arrow. Grabbing picks up gold if in same square. Releasing drops gold in same square. You bump into wall.
* **A:** Face, Move, Grab, Release, Shoot
* **S:** Perceive stench, breeze, glitter, bump (bump wall), & Wumpus scream when shot.

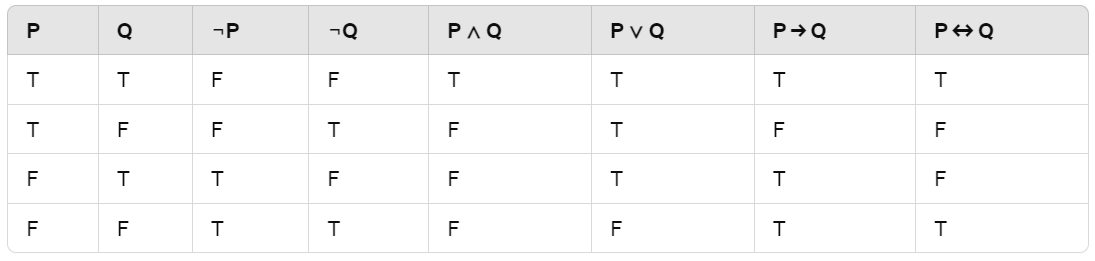
*Properties:*

* Partially observable: Agent can only perceive nearby rooms.
* Deterministic: Outcomes are known.
* Sequential: Order of actions matters.
* Static: Wumpus and pits don't move.
* Fully Observable: NO. The environment is divided into discrete rooms.
* Episodic: NO. Single-agent exploring the cave.



**Propositions**: declarative statements that are either true or false. Can be simple atomic or compound statements formed by combining simpler propositions using logical connectives.





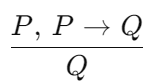
**Properties of Operators:**

1. *Commutativity:* P∧ Q= Q ∧ P or P ∨ Q = Q ∨ P.
2. *Associativity:* (P ∧ Q) ∧ R= P ∧ (Q ∧ R) (P ∨ Q) ∨ R= P ∨ (Q ∨ R)
3. *Identity element:* P ∧ True = P P ∨ True= True.
4. *Distributive:* P∧ (Q ∨ R) = (P ∧ Q) ∨ (P ∧ R). P ∨ (Q ∧ R) = (P ∨ Q) ∧ (P ∨ R).
5. *DE Morgan's Law:* ¬ (P ∧ Q) = (¬P) ∨ (¬Q) ¬ (P ∨ Q) = (¬ P) ∧ (¬Q).
6. *Double-negation elimination:* ¬ (¬P) = P.

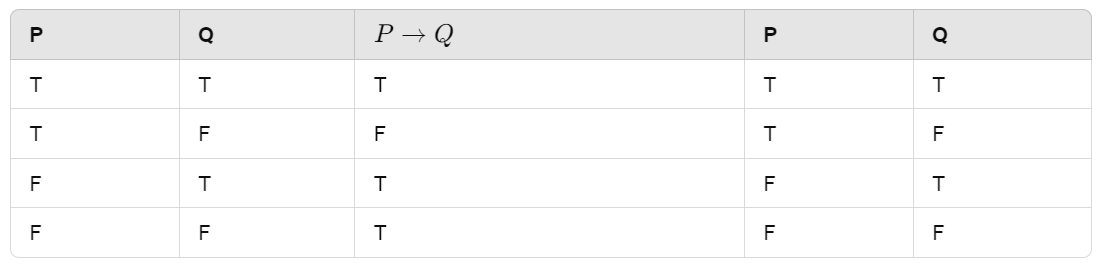
**Limitations:** Cannot express relationships like "all," "some," or "none." It cannot also represent statements in terms of their properties or logical reln beyond simple truth values.

**Inference rules** are formal logical structures used to derive conclusions from premises.

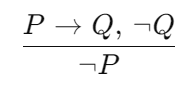
1. *Modus Ponens (Direct Inference)*

If P and P→Q are true, then Q is true.

Premise 1: If I am sleepy (P), then I go to bed (Q). Premise 2: I am sleepy (P).

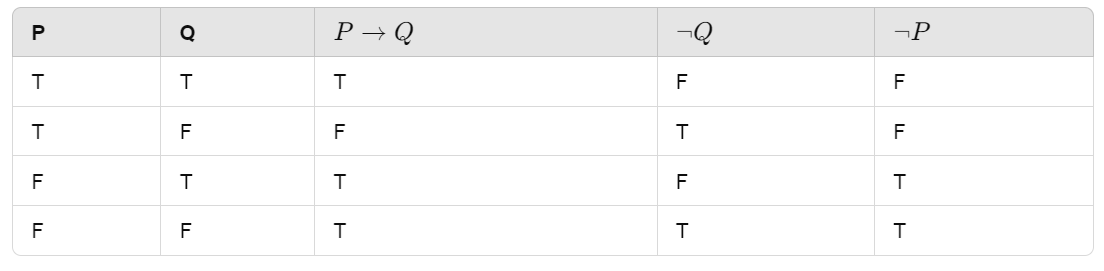
Conclusion: I go to bed (Q).

1. *Modus Tollens (Contrapositive Inference)*

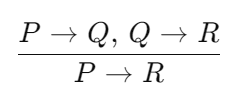
If P→Q and ¬Q are true, then ¬P is true.

Premise 1: If I am sleepy (P), then I go to bed (Q). Premise 2: I don’t go to bed (¬Q).

Conclusion: I am not sleepy (¬P).

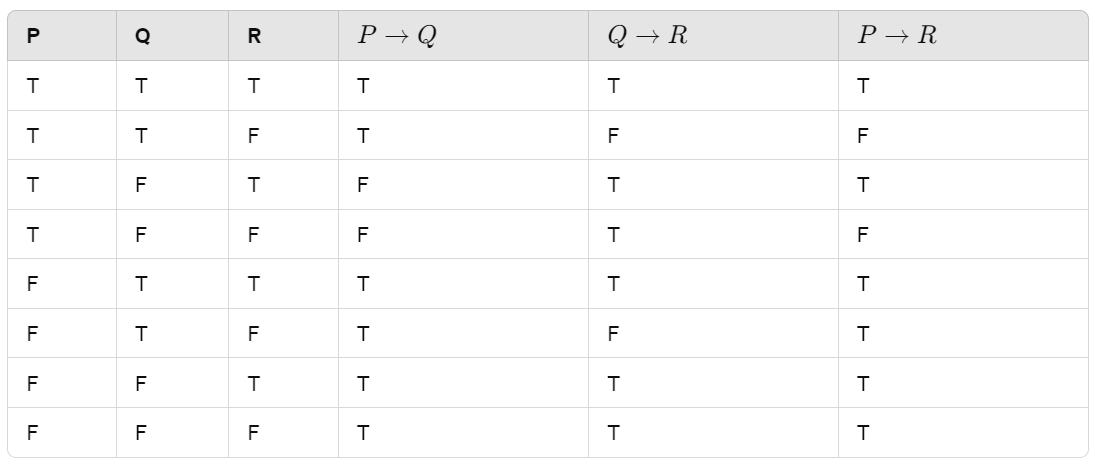


1. *Hypothetical Syllogism*

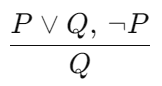
If P→Q and Q→R are true, then P→R is true.

Premise 1: If you have my home key (P), then you can unlock my home (Q).

Premise 2: If you can unlock my home (Q), then you can take my money (R).

Conclusion: If you have my home key (P), then you can take my money (R).

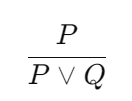
1. *Disjunctive Syllogism*

If P∨Q and ¬P are true, then Q is true.

Premise 1: Today is Sunday or Monday (P ∨ Q). Premise 2: Today isn’t Sunday (¬P).

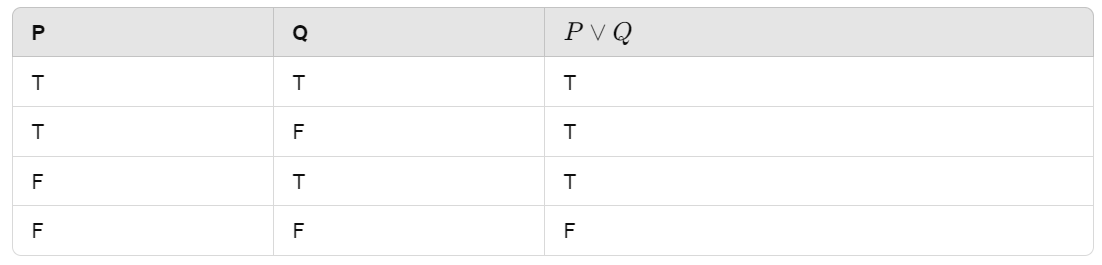
Conclusion: Today is Monday (Q).

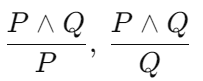


1. *Addition*

If P is true, then P∨Q is true.

Premise: I have a vanilla ice-cream (P).

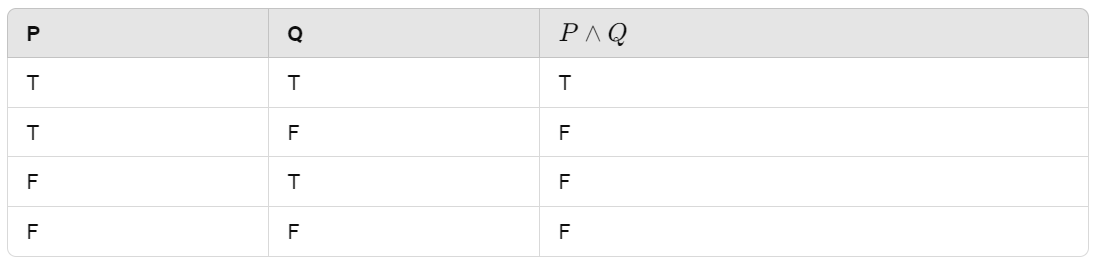
Conclusion: I have vanilla or chocolate ice-cream (P ∨ Q).

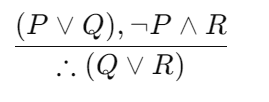
1. *Simplification*

If P∧Q is true, then P and Q are true.

Premise: It is raining and the street is wet (P ∧ Q).

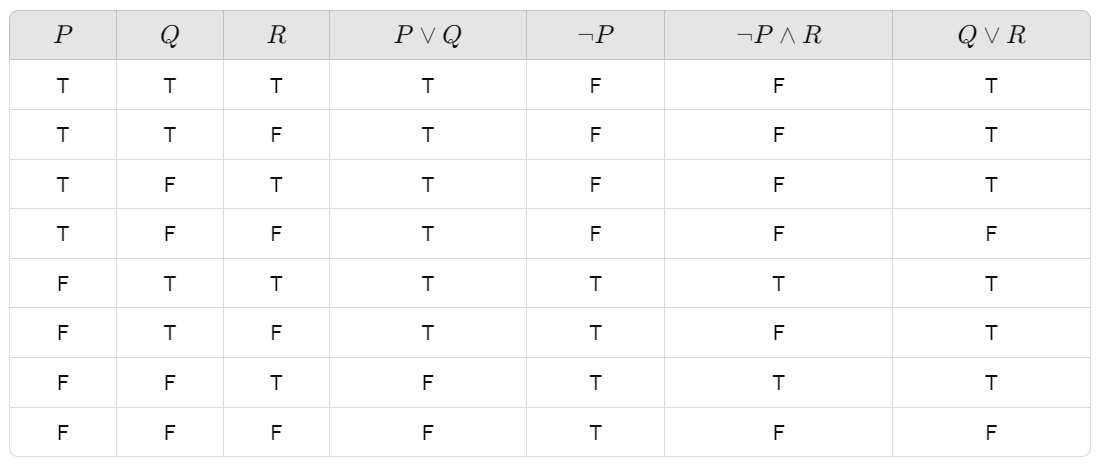
Conclusion: It is raining (P).



1. *Resolution*  
   If P∨Q and ¬P∧R are true, then Q∨R will also be true.  
   Statement 1: It is sunny or it is raining. (P∨Q)

Statement 2: It is not sunny and it is warm. (¬P∧R)

Conclusion: It is raining or it is warm. (Q∨R)



**Terminologies related to inference rules:**

1. *Implication:* P → Q. It is a Boolean expression.
2. *Converse:* The right-hand side goes to the left-hand side and vice-versa. Q → P.
3. *Contrapositive:* The negation of converse ¬ Q → ¬ P.
4. *Inverse:* The negation of implication ¬ P → ¬ Q.

**First-order logic (FOL)** (predicate logic). Expressive to represent natural lang statements in a concise & structured way. "Some humans are intelligent", "Sachin likes cricket."

1. *Constants:* Represent specific objects (e.g., 1, 2, John, Mumbai).
2. *Variables:* Represent unspecified objects (e.g., x, y, z).
3. *Predicates:* Express relationships between objects (e.g., Brother, Father).
4. *Functions:* Map objects to other objects (e.g., sqrt, LeftLegOf).
5. *Connectives:* Logical operators like ∧, ∨, ¬, ⇒, ⇔.
6. *Equality:* Symbolized by ==.
7. *Quantifiers:* ∀ (universal quantifier) and ∃ (existential quantifier).

**Atomic sentences:** predicate symbol followed by a sequence of terms enclosed in parentheses. Ravi and Ajay are brothers: Brothers(Ravi, Ajay). Chinky is a cat: Cat(Chinky).

**First-order logic statements**:

1. *Subject:* Subject is the main part of the statement.
2. *Predicate:* Relation, which binds two atoms together in a statement.

**Quantifiers in First-Order Logic:**

1. *Universal Quantifier (∀):* Represents statements that hold for all instances.

∀x Man(x) → Drink(x, Coffee). All men drink coffee.

1. *Existential Quantifier (∃):* Represents statements that hold for at least one instance.

∃x (Boys(x) ∧ Intelligent(x)). Some boys are intelligent.

**Properties of Quantifiers:**

1. ∀x∀y is similar to ∀y∀x.
2. ∃x∃y is similar to ∃y∃x.
3. ∃x∀y is not similar to ∀y∃x.

**Free and Bound Variables:**

1. *Free Variable:* Occurs outside the scope of the quantifier (e.g., z in ∀x ∃(y) P(x, y, z)).
2. *Bound Variable:* Occurs within scope of the quantifier (e.g., x and y in ∀x [A(x) B(y)]).

**Examples of FOL using quantifier:**

1. *All birds fly.*

Predicate is "fly(bird)." ∀x bird(x) →fly(x).

1. *Every man respects his parent.*

"respect(x, y)," where x=man, and y=parent. ∀x man(x) → respects (x, parent).

1. *Some boys play cricket.*

"play(x, y)," where x=boys, and y=game. ∃x boys(x) → play(x, cricket).

1. *Not all students like both Mathematics and Science.*

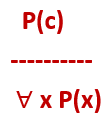
"like(x, y)," where x=student, and y=subject. ¬∀ (x) [ student(x) → like(x, Mathematics) ∧ like(x, Science)].

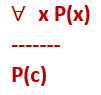
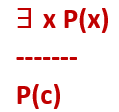
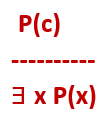
1. *Only one student failed in Mathematics.*

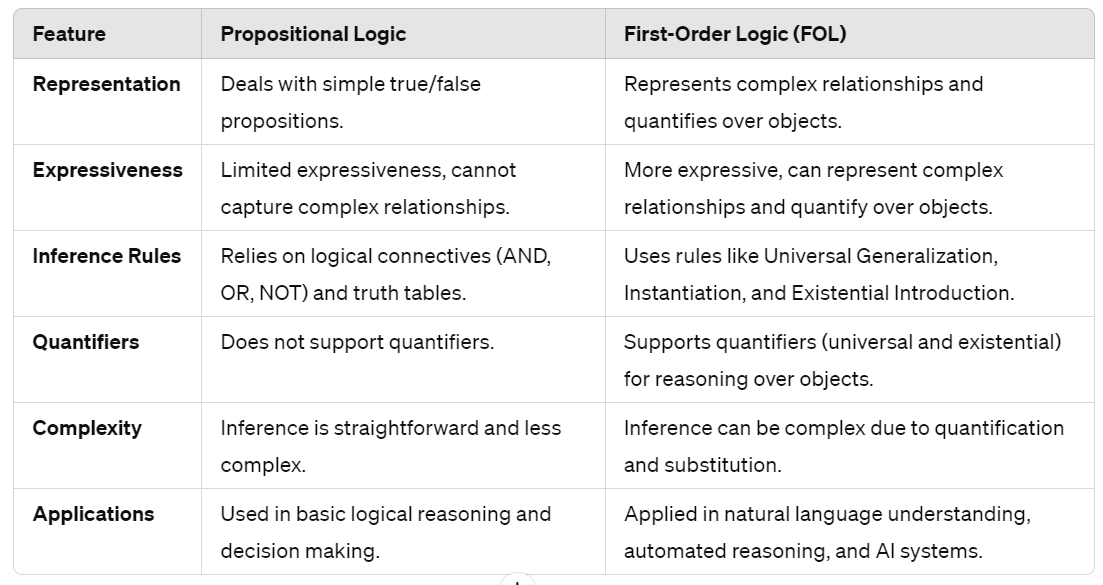
"failed(x, y)," where x= student, and y= subject. ∃(x) [ student(x) → failed (x, Mathematics) ∧∀ (y) [¬(x==y) ∧ student(y) → ¬failed (x, Mathematics)].

Inference in FOL is used to derive new facts or sentences from existing ones.

1. *Substitution:* Replacing terms or variables in formulas. If we have F[a/x], it means we're substituting the constant "a" for the variable "x".
2. *Equality:* Denote that two terms refer to the same object. For example, Brother(John) = Smith implies that the object referred to by Brother(John) is the same as that referred to by Smith. It can also be used in negated form (e.g., x ≠ y).

**FOL Inference Rules for Quantifiers:**

1. *Universal Generalization:* If premise P(c) is true for any arbitrary element c in the universe of discourse, then we can conclude ∀x P(x). P(c): "A byte contains 8 bits", so for ∀ x P(x) "All bytes contain 8 bits.", it will also be true.
2. *Universal Instantiation:* Allows us to infer any sentence P(c) by substituting a ground term (constant) c from ∀x P(x). From "Every person likes ice-cream" (∀x P(x)), we can infer "John likes ice-cream" (P(c)).
3. *Existential Instantiation:* Existential Elimination, it allows us to replace an existential sentence ∃x P(x) with P(c) for a new constant symbol c. From ∃x Crown(x) ∧ OnHead(x, John), we infer Crown(K) ∧ OnHead(K, John), where K is a new constant.
4. *Existential Introduction:* Existential Generalization, it allows us to infer ∃x P(x) if there exists an element in the universe with property P. "Priyanka got good marks in English" implies "Someone got good marks in English".



**Unification** is the process of making two different logical atomic expressions identical by finding a substitution. Used to find the most general unifier (MGU) for two expressions, which represents the most general substitution that makes them identical.

Predicate symbol must be same. Number of Arguments in both expressions must be identical. Unification will fail if there are two similar variables present in the same expression.

Let Ψ1 and Ψ2 be two atomic sentences and 𝜎 be a unifier such that, Ψ1𝜎 = Ψ2𝜎, then it can be expressed as UNIFY(Ψ1, Ψ2).

Example: Find the MGU for Unify{King(x), King(John)}

Let Ψ1 = King(x), Ψ2 = King(John), Substitution θ = {John/x} is a unifier for these atoms and applying this substitution, and both expressions will be identical.

***1. Find the MGU of {p(f(a), g(Y)) and p(X, X)}***

S0 => Here, Ψ1 = p(f(a), g(Y)), and Ψ2 = p(X, X)

SUBST θ= {f(a) / X}

S1 => Ψ1 = p(f(a), g(Y)), and Ψ2 = p(f(a), f(a))

SUBST θ= {f(a) / g(y)}, Unification failed.

***2. Find the MGU of {p(b, X, f(g(Z))) and p(Z, f(Y), f(Y))}***

Here, Ψ1 = p(b, X, f(g(Z))) , and Ψ2 = p(Z, f(Y), f(Y))

S0 => { p(b, X, f(g(Z))); p(Z, f(Y), f(Y))}

SUBST θ={b/Z}

S1 => { p(b, X, f(g(b))); p(b, f(Y), f(Y))}

SUBST θ={f(Y) /X}

S2 => { p(b, f(Y), f(g(b))); p(b, f(Y), f(Y))}

SUBST θ= {g(b) /Y}

S2 => { p(b, f(g(b)), f(g(b)); p(b, f(g(b)), f(g(b))} Unified Successfully.

And Unifier = { b/Z, f(Y) /X , g(b) /Y}.

***3. Find the MGU of {p (X, X), and p (Z, f(Z))}***

Here, Ψ1 = {p (X, X), and Ψ2 = p (Z, f(Z))

S0 => {p (X, X), p (Z, f(Z))}

SUBST θ= {X/Z}

S1 => {p (Z, Z), p (Z, f(Z))}

SUBST θ= {f(Z) / Z}, Unification Failed.

Hence, unification is not possible for these expressions.

***4. Find the MGU of UNIFY(prime (11), prime(y))***

Here, Ψ1 = {prime(11) , and Ψ2 = prime(y)}

S0 => {prime(11) , prime(y)}

SUBST θ= {11/y}

S1 => {prime(11) , prime(11)} , Successfully unified.

Unifier: {11/y}.

***5. Find the MGU of Q(a, g(x, a), f(y)), Q(a, g(f(b), a), x)}***

Here, Ψ1 = Q(a, g(x, a), f(y)), and Ψ2 = Q(a, g(f(b), a), x)

S0 => {Q(a, g(x, a), f(y)); Q(a, g(f(b), a), x)}

SUBST θ= {f(b)/x}

S1 => {Q(a, g(f(b), a), f(y)); Q(a, g(f(b), a), f(b))}

SUBST θ= {b/y}

S1 => {Q(a, g(f(b), a), f(b)); Q(a, g(f(b), a), f(b))}, Successfully Unified.

Unifier: [a/a, f(b)/x, b/y].

***6. UNIFY(knows(Richard, x), knows(Richard, John))***

Here, Ψ1 = knows(Richard, x), and Ψ2 = knows(Richard, John)

S0 => { knows(Richard, x); knows(Richard, John)}

SUBST θ= {John/x}

S1 => { knows(Richard, John); knows(Richard, John)}, Successfully Unified.

Unifier: {John/x}.

**Lifting** refers to the process of extending an inference rule or operation from one domain or level of abstraction to another, typically from a lower level to a higher level. It allows for the generalization of inference rules and operations to handle more complex logical structures, such as quantification, functions, and higher-order predicates. Moving from propositional logic to FOL involves lifting inference rules to handle quantification and predicates.

**Resolution**: theorem proving technique in logic, used for proving statements by contradiction. We can resolve two clauses which are given below:

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

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

They can be unified with unifier θ= [a/f(x), and b/x], & it’ll generate a resolvent clause:

[Animal (g(x) V ￢ Kills(f(x), x)].

**Steps for Resolution:**

1. *Conversion to FOL:* Convert given statements into First-Order Logic.
2. *Conversion to CNF:* Represent FOL statements as a conjunction of clauses.
3. *Negation:* Negate the statement to be proven (proof by contradiction).
4. *Resolution Graph:* Draw a resolution graph, performing unification with substitution.

**Example:**

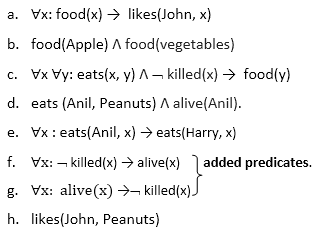
John likes all kind of food. Apple and vegetable are food

Anything anyone eats and not killed is food. Anil eats peanuts and still alive.

Harry eats everything that Anil eats.

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

***Step-1: Conversion of Facts into FOL***



***Step-2: Conversion of FOL into CNF***

***Eliminate all implication (→), move negation (¬)inwards and rewrite***

∀x ¬ food(x) V likes(John, x)

food(Apple) Λ food(vegetables)

∀x ∀y ¬ eats(x, y) V killed(x) V food(y)

eats (Anil, Peanuts) Λ alive(Anil)

∀x ¬ eats(Anil, x) V eats(Harry, x)

∀x ¬killed(x) ] V alive(x)

∀x ¬ alive(x) V ¬ killed(x)

likes(John, Peanuts).

***Rename variables or standardize variables***

∀x ¬ food(x) V likes(John, x)

food(Apple) Λ food(vegetables)

∀y ∀z ¬ eats(y, z) V killed(y) V food(z)

eats (Anil, Peanuts) Λ alive(Anil)

∀w¬ eats(Anil, w) V eats(Harry, w)

∀g ¬killed(g) ] V alive(g)

∀k ¬ alive(k) V ¬ killed(k)

likes(John, Peanuts).

***Drop Universal quantifiers.***

¬ food(x) V likes(John, x)

food(Apple)

food(vegetables)

¬ eats(y, z) V killed(y) V food(z)

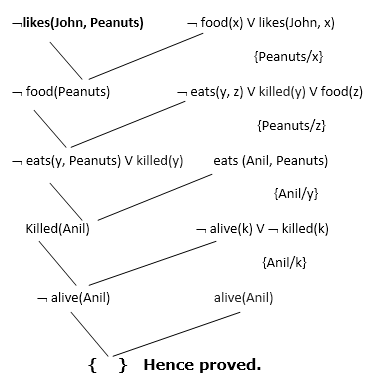
eats (Anil, Peanuts)

alive(Anil)

¬ eats(Anil, w) V eats(Harry, w)

killed(g) V alive(g)

¬ alive(k) V ¬ killed(k)

likes(John, Peanuts).

***Step-3: Negate the statement to be proved***

We will apply negation to conclusion statements, written as ¬likes(John, Peanuts)

***Step-4: Draw Resolution graph:***



***Example:***

"As per the law, it is a crime for an American to sell weapons to hostile nations. Country A, an enemy of America, has some missiles, and all the missiles were sold to it by Robert, who is an American citizen."

***Prove that "Robert is criminal."***

***Facts Conversion into FOL:***

It is a crime for an American to sell weapons to hostile nations.

American (p) ∧ weapon(q) ∧ sells (p, q, r) ∧ hostile(r) → Criminal(p) ...(1)

Country A has some missiles. ?p Owns(A, p) ∧ Missile(p).

Owns(A, T1) ......(2)

Missile(T1) .......(3)

All of the missiles were sold to country A by Robert.

?p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A) ......(4)

Missiles are weapons. Missile(p) → Weapons (p) .......(5)

Enemy of America is known as hostile. Enemy(p, America) →Hostile(p) ........(6)

Country A is an enemy of America. Enemy (A, America) .........(7)

Robert is American. American(Robert). ..........(8)

***In forward-chaining, In backward-chaining***,