[redacted]

Unification

→ It is all about making the expression looks identical so, for the given expression to make them look similar, we need to do substitution –

Ex: [P[x, F/y)] , [P(a, F/g/z)].

x = a
y = g/z

Unification [a/x, g/z/y].

→ Ist expression will be identical to the second expression & the substitution set will be [a/x, g(z)/y]

→ Conditions for Unification:

1) Predicate symbol must be same atoms of expression. with diff predicate symbol can be never unified.
2) No. of arguement in both expression must be identical
3) Unification will fail if there are 2 similar variables present in same expression.

→ Unification Algorithm:

Unify (L_1, L_2)

If L_1 or L_2 is a variable or constant then:

(a) If L_1 & L_2 are identical return NIL

(b) Else if L_1 is a variable, then if L_1 occurs in L_2 then return Fail else return {(L_2|L_1)}

(c) Else if L_2 is a variable, then if L_2 occurs in L_1, then return Fail, else return {(L_1|L_2)}.

(d)

RESOLUTION.

* It is a theorem proving technique that proof by help of contradiction. It is used, if there are various statement are given and need to prove a conclusion of those statements.
→ Unification is a key concept in proof by resolution
→ Resolution is a single inference rule which can efficiently operate on Conjunctive Normal Form or clausal Form.
→ Clause : Disjunctive of literals is called as clause.
→ CNF : A sentence represented as a conjunction of clause is said to be CNF.

* <u>Steps for Resolution:</u>
1) conversion of facts into FOL.
2) convert FOL stmt. into CNF.
3) Negate the stmt which needs to be prove by contra.
4) Draw the resolution graph (unification)

[Facts (stmts) raw → FOL (quantifiers) universal → CNF → Negate stmt → Resolution Graph.]

<u>Example:</u>
a) John likes all kind of food.
b) Apple & vegetable are food.
c) Anything anyone eats & not killed is food.
d) Anil eats peanuts & is still alive
e) Harry eats everything that Anil eats.
f) John likes peanuts.

<u>Step 1 :</u> (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) $eats(Anil, peanuts) \wedge alive(Anil)$

} Facts into FOL

(e) $\forall x$: eats (Anil, x) → eats (Harry, x)
(f) $\forall x$ [killed (x)] → alive (x) } added predicates.
(g) $\forall x$ alive (x) → ¬ killed (x) }
(h) likes (John, peanuts)

Step 2: conversion of FOL into CNF:
↳ makes easier for resolution proof

i) Eliminate all implications (→) and rewrite the stmts.
Follow: ∴ a → b = ¬(a ∨ b)

(a) $\forall x$ ¬ food (x) ∨ likes (John, x)
(b) Food (Apple) ∧ food (vegetables) No implies ; No change.
(c) $\forall x$ $\forall y$ ¬[eats(x, y) ∧ ¬ killed (x)] ∨ food (y).
(d) eats (Anil, peanuts) ∧ alive (Anil)
(e) $\forall x$ ¬ eats (Anil, x) ∨ eats (Harry, x)
(f) $\forall x$ ¬ [¬ killed (x)] ∨ alive (x).
(g) $\forall z$ ¬ alive (x) ∨ ¬ killed (x)
(h) ¬ likes (John, peanuts)

(ii) Move negation inwards & re-write again.

a) $\forall x$ ¬ food (x) ∨ likes (John, x)
b) Food (apple) ∧ food (vegetable)
c) $\forall x$ $\forall y$ ¬ eats(x, y) ∨ killed (x)
d) eats (Anil, peanuts) ∧
) $\forall x$ ¬ eats (Anil, x) ∨ eats (Harry, x)
) $\forall x$ ¬ killed (x) ∨ alive (x)
) $\forall z$ ¬ alive (x) ∨ ¬ killed (x)
) likes (John, peanuts)

i) Rename the variables / standardize variables
) $\forall x$ ¬ food (x) ∨ likes (John, x)
) food (apple) ∧ food (vegetables)
) $\forall y$ $\forall z$ (Anil, w) ∧ alive (Anil)
eats (Anil, peanuts ∧ alive (Anil)

Resolution - contd. ~~[illegible]~~

iv) Eliminate esixtential quantifier.

→ But here it contains ∀ quantifier so all the remaining statements may take place.

v) Drop Universal Quantifier:

a) ~ food (x) V likes (John, x)
b) food (Apple)
c) Food (vegetable)
d) ~ eats (y, z) V killed (y) V food (z)
e) eats (Anil, peanuts)
f) alive (Anil)
g) ~ eats (Anil, w) V eats (Harry, w)
h) killed (g) V alive (g).
i) ~ alive (k) V ~ killed (k)
j) like (John, peanuts).

Step3: Negate the stmts proved:

In this stmt we will apply negation to the conclusion stmt which will likes as ~ like (John, peanuts)

Step 4: Draw Resolution Graph.

Now, in this we'll solve the pblm by resolution tree using substitution for all the above problem it'll be given as:

Probabilistic Reasoning

→ Uncertainity
→ Unpredictable } outcomes.
→ Comes from laziness & ignorance.
→ using logic & probability to handle uncertainity.
→ Probability based reasoning is something opposite from Boolean
→ Ex: Doctor examines a patient, his history, symptoms, Based on the test result will be analysed.
 – 90% of disease will be cured if diagnosed properly.
Rarely, it may also happen at times if a doctor by mistake missed any test which was one of the important ones for the diagnosis purpose, then the results will be affected drastically.

→ Sources that causes Uncertainity:

1) Information is obtained personally.
2) experimental error.
3) Random event occurs in major event
4) Equipment fault
5) Temperature variation or climatic changes occurs

Scanned by TapScanner

Utility Theory

→ Utility Function & Rational Agent

These are 2 Basis of utility theory - it is depended on how a utility fn. is applied to get the maximum output / result from the available data.

- They expect to behave like an ideal agent that will provide an ideal result, which is infeasible in practical situation.
- So, this Rational Agent is not possible in real world scenario but it is possible to be in near by situation i.e. will behave like one and try to use these utility function and work like a rational agent.

→ Let us assume that: Agent (A) comes at state (S)

S → action 1 - Marks < 33 - fail

S → action 2 - marks = 75 - Good.

S → action 3. (Check did it receive 100/100 or not)

- If marks = 100 ; then the system'll store in it that it'll perform only this action while coming to this state.
- But if in case marks = 63 ; then it'll decrease the rank by 1 and whenever it'll come next time at same state S.

→ It will try to achieve a better mark scenario; so it'll perform another action to achieve best outcome or it'll try to pufoom the 2nd action again so as to get atleast 75 marks which was last best they achieved.

→ It would depend on the training on the machine as it'll try to perform the action giving the best outcome or ideal one nearly.

→ It provides a Rank to all the actions with the help of utility function and then they decide which action has to be performed.

→ In utility fn. they apply some formula to evaluate the value of fn. and then changes it not achieved ideal result or expected one.

→ So, in this way an Agent implement an Utility Theory using utility function and try to behave like a Rational Agent.

→ It'll try to improve himself in each training so as to achieve a better outcome every time.