11/10/25

LAB 7

Implement unification in First Order Logic

Step1: If Ψ_1 or Ψ_2 is a variable or constant, then:

a) If Ψ_1 or Ψ_2 are identical, then return NIL

b) Else if Ψ_1 is a variable,

a. Then if Ψ_1 occurs in Ψ_2, then return FAILURE

b. Else return $\{(\Psi_2 / \Psi_1)\}$.

c) Else if Ψ_2 is a variable,

a. If Ψ_2 occurs in Ψ_1 then return FAILURE,

b. Else return $\{(\Psi_1 / \Psi_2)\}$

d) Else return FAILURE

Step2: If the initial Predicate symbol in Ψ_1 and Ψ_2 are not the same, then return FAILURE.

Step3: If Ψ_1 and Ψ_2 have a different number of arguments, then return FAILURE

Step4: Set Substitution set (SUBST) to NIL

Step5: For i=1 to the number of elements in Ψ_1,

a) Call Unify function with the ith element of Ψ_1 and ith element of Ψ_2 and put the result into S

b) If S = failure then return failure

c) If S ≠ NIL then do,

a. Apply S to the remainder of both L_1 & L_2

b. SUBST = APPEND(S, SUBST)

Step6: Return SUBST

Output :-

Question : $S = \{ q(x, f(Y), z) ; q(f(a), f(b), f(b)) \}$

Step 1 : subst $\theta = \{ x / f(a) \}$

$S = \{ q\, f(a), f(Y), z) ; q(f(a), f(b), f(b)) \}$

Step 2 :- subst $\theta = \{ Y / b \}$

$S = \{ q\, f(a), f(b), z) ; q(f(a), f(b), f(b)) \}$

Step 3 :- subst $\theta = \{ z / f(b) \}$

$S = \{ q(f(a), f(b), f(b)) ; q(f(a), f(b), f(b)) \}$

$S = \{ q(f(a), f(b), f(b)) ; q(f(a), f(b), f(b)) \}$

$\theta = \{ x / f(a), Y / b, z / f(b) \}$

(Unified Successfully)

Forward Chaining Using FOL

function FOL-FC-ASK(KB, a) returns a substitution or false

inputs: KB, a knowledge base, a set of first-order definite clauses a, the query, an atomic sentence.

local variables: new, the new sentences inferred on each iteration

repeat until new is empty

new ← { }

for each rule in KB do

$(p_1 \wedge \ldots \wedge p_n \Rightarrow q) \leftarrow$ STANDARDIZE-VARIABLES(rule)

for each θ such that SUBST($\theta, p_1 \wedge \ldots \wedge p_n$) = SUBST($\theta, p'_1 \wedge \ldots \wedge p'_n$)

for some $p'_1, \ldots, p'_n$ in KB

$q' \leftarrow$ SUBST(θ, q)

if q' does not unify with some sentence already in KB or new then

add q' to new

$\phi \leftarrow$ UNIFY(q', a)

if ϕ is not fail then return ϕ

add new to KB

return false

Output :-

Initial facts :

minister (Birbal)
ruler (Akbar)

-- Forward Chaining Steps --

Derived king(Akbar) using ruler(x) → king(x)

Derived advisor(Birbal) using minister(x) → advisor(x)

Derived powerful (Akbar) using king(x) → powerful(x)

Final facts :

powerful (Akbar)
minister (Birbal)
advisor (Birbal)
ruler (Akbar)
king(Akbar)

Goal powerful (Akbar) reached !

14/11/25