

Implement unification in first order logic

Step 1: If ψ 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 $E(\psi_2/\psi_1)$

(c) Else if ψ_2 is a variable

(a) If ψ_2 occurs in ψ_1 , then return FAILURE

(b) Else return $E(\psi_1/\psi_2)$

(d) Else return FAILURE

Step 2: If the initial predicate symbol in ψ_1 and ψ_2 are not same, then return FAILURE

Step 3: If ψ_1 & ψ_2 have a diff no. of arguments, then return FAILURE

Step 4: Set Substitution Set (SUBST) to NIL

Step 5: For $i=1$ to the no. of elements in ψ_1

(a) Call unify function with the i^{th} element of ψ_1 & i^{th} element of ψ_2 , & put the result into S

(b) If S = failure then return failure

(c) If $S \neq \text{failure}$ then do

(a) Apply S to the remainder of both ψ_1 & ψ_2

(b) SUBST = APPEND(S, SUBST)

Step 6: Return SUBST

15
Q1) Check an ex. to run :-
Team 1: ('knows', 'John', ('person', 'Alice'))
Team 2: ('knows', 'John', ('person', 'Bob'))
Unification succeeds.

Enter 1 as 2:

Unifying frame:

Team 1: ('knows', 'John', ('person', 'Alice'))
Team 2: ('knows', ('person', 'Bob'))

Unification failed

Jan