19D03187
Radhakrishna

6. unification is a process of making two different logical atomic expressions identical by finding a substitution

unification depends on the substitution process. It takes two literals as input and makes them identical using substitution.

Let ψ_1 and ψ_2 be two atomic sentences and σ be a unifier such that $\psi_1\sigma = \psi_2\sigma$, then it can be expressed as

unify (ψ_1, ψ_2)

Example: Find the MGU for unify{(king(x), king(John)}

Let ψ_1 = king(x),
ψ_2 = king(John)

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

The UNIFY Algorithm is used for unification, which takes two atomic sentences and returns a unifier for those sentences (if any exist)

unification is a key component of all first order inference algorithms

It returns fail if the expressions do not match with each other.

The substitution variables are called Most General unifier (or MGU.

unification Algorithm:-

Algorithm: unify (ψ_1, ψ_2)

step-1 :- 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

~~(If ψ_2 occurs in ψ_1 then)~~

b. Else return {(ψ_2/ψ_1)}

c) Else if ψ_2 is a variable,

a. If ψ_2 occurs in ψ_1 then return FAILURE

b. Else return {(ψ_1/ψ_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 and ψ_2 have a different 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 ith element of ψ_1 and ith element of ψ_2 and put the result into S.

b) If S = failure then returns FAILURE

c) If S ≠ NIL then do

a. Apply S to the remainder of both L1 and L2

b. SUBST = APPEND (S, SUBST)

step-6 :- Return SUBST

Example

→ UNIFY (knows (Richard, x), knows (Richard, John))

Here Ψ_1 = knows(Richard, x)

Ψ_2 = knows (Richard, John)

S_0 ⇒ { knows (Richard, x) ; knows (Richard, John)}

SUBST θ = { John(x }

S_1 ⇒ { knows(Richard, John) ; knows (Richard, John)}

Successfully unified

Unifier = { John / x }

→ { P (x, x) and P (z, f(z)) }

Here Ψ_1 = { P(x, x) }

Ψ_2 = P (z, f(z))

So ⇒ {P (x, x) , P(z, f(z)) }

SUBST θ = { x / z }

S_1 ⇒ {P(z, z), P (z, f(z))}

SUST θ = { f(z) / z }, unification failed.