Lab-6

Date 22/11/2024
Page

⇒ Implement unification in FOL

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 NULL.
b) Else if Ψ_1 is a variable,
a. then if Ψ_1 occurs in Ψ_2, then return Fail
b. Else return $\{(\Psi_2/\Psi_1)\}$
c. Else if Ψ_2 is variable
i. if Ψ_2 occurs in Ψ_1 return Fail
ii. Else return $\{(\Psi_1/\Psi_2)\}$
d. Else return Fail.

Step 2: If the initial Predicate symbol in Ψ_1 and Ψ_2 are not same, return Fail

Step 3: If Ψ_1 and Ψ_2 have a different no. of arguments then return Fail

Step 4: Set substitution set (SUBST) to NULL

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 result into S
b) If S= Failure return Failure
c) If S ≠ NULL then do
a. apply S to remaining of both L1 and L2
b. subst = Append (S, subst)

Step 6: Return SUBST

Date __/__/__
Page

Outputs:

S1 = ('P', 'x', ('J', 'x'), ('y'))
S2 = ('P', 'a', 'y', ('J', 'x'))

Unification successful.
Substitution : {'x' : 'a', 'y' : ('J', 'x')}

S2[u/z]

stfpurcas

December 21, 2024

[1]:
```
print("Name:Sudarshan Komar","USN:1BM22CS291",sep="\n")

def unify(s1, s2, theta={}):
    """
    Unifies two first-order logic expressions.

    Args:
        s1: The first expression.
        s2: The second expression.
        theta: The current substitution.

    Returns:
        A substitution that unifies s1 and s2, or None if they cannot be␣
 ↪unified.
    """

    if theta is None:
        return None

    if s1 == s2:
        return theta

    if isinstance(s1, str) and s1.islower():  # Variable
        return unify_var(s1, s2, theta)

    if isinstance(s2, str) and s2.islower():  # Variable
        return unify_var(s2, s1, theta)

    if isinstance(s1, tuple) and isinstance(s2, tuple) and len(s1) == len(s2):
        return unify(s1[1:], s2[1:], unify(s1[0], s2[0], theta))

    return None

def unify_var(var, x, theta):
    if var in theta:
        return unify(theta[var], x, theta)
    elif x in theta:
```

```
        return unify(var, theta[x], theta)
    elif occurs_check(var, x, theta):
        return None  # Occurs check failed
    else:
        theta[var] = x
        return theta

def occurs_check(var, x, theta):
    if var == x:
        return True
    elif isinstance(x, str) and x.islower() and x in theta:
        return occurs_check(var, theta[x], theta)
    elif isinstance(x, tuple):
        for arg in x:
            if occurs_check(var, arg, theta):
                return True
    return False


s1 = ('p', 'x', ('f', 'x'), ('y'))
s2 = ('p', 'a', 'y', ('f', 'x'))

substitution = unify(s1, s2)

if substitution:
    print("Unification successful:")
    print(f"Substitution: {substitution}")
else:
    print("Unification failed.")
```

```
Name:Sudarshan Komar
USN:1BM22CS291
Unification successful:
Substitution: {'x': 'a', 'y': ('f', 'x')}
```