

Implement unification in first order logic.

Algorithm:

Unify( $E_1, E_2$ ):

Step 1: If  $E_1$  or  $E_2$  is a variable or constant, then:

a) If  $E_1$  or  $E_2$  are identical, then return NIL.

b) Else if  $E$  is a variable.

a. then if  $E_1$  occurs in  $E_2$ , then return FAILURE

b. Else return  $\{(E_2 / E_1)\}$ .

c) Else if  $E_2$  is a variable;

a. if  $E_2$  occurs in  $E_1$ , then return FAILURE

b. Else return  $\{(E_1 / E_2)\}$

d) Else return FAILURE

Step 2: If the initial Predicate symbol in  $E_1$  and  $E_2$  are not same, then return FAILURE.

Step 3: If  $E_1$  and  $E_2$  have different number of arguments, then return FAILURE.

Step 4: Set Substitution set (SUBST) to NIL.

Step 5: For  $i=1$  to the number of elements in  $E_1$

a) Call Unify function with the  $i$ th element of  $E_1$  and  $i$ th element of  $E_2$ , and put the result into  $s$ .

b) if  $s$ : failure then returns Failure.

c) if  $s \neq \text{NIL}$  then do,

a. Apply  $s$  to the remainder of both  $E_1$  &  $E_2$

b. SUBST = APPEND( $s$ , SUBST)

Step 6: Return SUBST.

Output:

eg1:

Enter E1:  $P(A(), x)$

Enter E2:  $P(G(), y)$

unification failed!

eg2:

Enter E1:  $P(A(), x)$

Enter E2:  $P(A(), y)$

unification successful

Substitutions

$x \rightarrow y$