

## Unification in first order logic

Algo:

Step 1: If  $\psi_1$  or  $\psi_2$  is a variable or constant, then:

a) If  $\psi_1$  and  $\psi_2$  are identical, then return NIL.

b) Else if  $\psi_1$  is a variable,

a) then if  $\psi_1$  occurs in  $\psi_2$ , then  
return FAILURE

b) Else return  $\{(\psi_2/\psi_1)\}$ .

c) Else if  $\psi_2$  is a variable,

a) If  $\psi_2$  occurs in  $\psi_1$ , then return  
FAILURE

b) Else return  $\{(\psi_1/\psi_2)\}$

d) Else return FAILURE.

Step 2: If the initial Predicate symbol in  $\psi_1$   
and  $\psi_2$  are not same, then return FAILURE

Step 3: If  $\psi_1$  and  $\psi_2$  have a different number  
of arguments, then return FAILURE.

Step 4: Set Substitution set (SUBT) to NIL

a) Call unify function with the  $i$ th element  
of  $\psi_1$  and  $i$ th element of  $\psi_2$  and put  
the result into S.

b) If  $S = \text{failure}$  then return failure

- c) If  $S \neq \text{NIL}$  then do,
  - Apply  $S$  to remainder of both  $L1$  and  $L2$

b) If  $\text{SUBST} = \text{APPEND}(S, \text{SUBST})$ .

Step 6: Return  $\text{SUBST}$ .

OUTPUT:

choose an example to run:

1. Example where unification fails
2. Example where unification succeeds

Enter 1 or 2: 2

unifying terms:

Term 1: ('knows', 'John', ('person', 'Alice'))

Term 2: ('knows', 'John', ('person', 'Bob'))

Unification succeeds.

Enter 1 or 2: 1

unifying terms:

Term 1: ('knows', 'John', ('person', 'Alice'))

Term 2: ('knows', ('person', 'Bob'),

Unification failed.