

## Un Algoritmo di Unificazione

```
FUNCTION UNIFY (t1, t2: termine; sigma: sostituzione): sostituzione;  
VAR a, b: temine;  
BEGIN  
IF sigma = false THEN UNIFY := false  
ELSE  
  BEGIN  
    a := SOST(sigma, t1); b := SOST(sigma, t2);  
    IF a e b sono costanti e a = b THEN UNIFY := sigma  
    ELSE  
      IF a e` una variabile non presente in b THEN    (* occur check *)  
        aggiorna sigma con la nuova sost. {a/b}; UNIFY := sigma  
      ELSE  
        IF b e` una variabile non presente in a THEN    (* occur check *)  
          aggiorna sigma con la nuova sost. {b/a}; UNIFY := sigma  
        ELSE  
          IF a e b sono termini composti THEN  
            UNIFY( testa(a), testa(b), UNIFY(coda(a), coda(b), sigma) )  
          ELSE  
            UNIFY := false  
          END  
        END  
      END  
    END  
  END  
END.  
END.
```

Un termine composto e` rappresentato con una lista:  $(f A (g B)) = f(A, g(B))$   
 $\text{testa}(L) =$  primo elemento della lista  $L$ .  $\text{coda}(L) =$  resto di  $L$ .