

# 9

## INFERENCE IN FIRST-ORDER LOGIC

**function** UNIFY( $x, y, \theta$ ) **returns** a substitution to make  $x$  and  $y$  identical

**inputs:**  $x$ , a variable, constant, list, or compound expression  
 $y$ , a variable, constant, list, or compound expression  
 $\theta$ , the substitution built up so far (optional, defaults to empty)

**if**  $\theta = \text{failure}$  **then return** failure

**else if**  $x = y$  **then return**  $\theta$

**else if** VARIABLE?( $x$ ) **then return** UNIFY-VAR( $x, y, \theta$ )

**else if** VARIABLE?( $y$ ) **then return** UNIFY-VAR( $y, x, \theta$ )

**else if** COMPOUND?( $x$ ) **and** COMPOUND?( $y$ ) **then**

**return** UNIFY( $x$ .ARGS,  $y$ .ARGS, UNIFY( $x$ .OP,  $y$ .OP,  $\theta$ ))

**else if** LIST?( $x$ ) **and** LIST?( $y$ ) **then**

**return** UNIFY( $x$ .REST,  $y$ .REST, UNIFY( $x$ .FIRST,  $y$ .FIRST,  $\theta$ ))

**else return** failure

---

**function** UNIFY-VAR( $var, x, \theta$ ) **returns** a substitution

**if**  $\{var/val\} \in \theta$  **then return** UNIFY( $val, x, \theta$ )

**else if**  $\{x/val\} \in \theta$  **then return** UNIFY( $var, val, \theta$ )

**else if** OCCUR-CHECK?( $var, x$ ) **then return** failure

**else return** add  $\{var/x\}$  to  $\theta$

**Figure 9.1** The unification algorithm. The algorithm works by comparing the structures of the inputs, element by element. The substitution  $\theta$  that is the argument to UNIFY is built up along the way and is used to make sure that later comparisons are consistent with bindings that were established earlier. In a compound expression such as  $F(A, B)$ , the OP field picks out the function symbol  $F$  and the ARGS field picks out the argument list  $(A, B)$ .