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 $\theta$ = 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 θ 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).