**Unit II CS6659 Artificial Intelligence**

**Unification**

1. The algorithm works by comparing the structures of the inputs, element by element.
2. The substitution Ѳ 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.
3. In a compound expression, such as F(A, B), the function OP picks out the function symbol F and the function ARCS picks out the argument list (A, B).
4. An algorithm for computing most general unifiers is shown in **Figure 1.**
5. The process is very simple: recursively explore the two expressions simultaneously "side by side," building up a unifier along the way, but failing if two corresponding points in the structures do not match.
6. There is one expensive step: when matching a variable against a complex term, one must check whether the variable itself occurs inside the term; if it does, the match fails OCCURCHECK because no consistent unifier can be constructed.
7. This so-called **occur check** makes the complexity of the entire algorithm quadratic in the size of the expressions being unified.

****

**Figure 1:** The unification algorithm.