**NAME:** NIHA

**USN:** 1BM18CS060

**DATE:** 04-12-2020

**LAB 4:** Implement unification in first order logic

**Explanation and Rules for Unification:**

Unification: The process of finding all legal substitutions that make logical expressions look identical.

Unification is a pattern matching procedure that takes two atomic sentences, called literals, as input, and returns failure if they do not match and a substitution list, Theta, if they do match.

Theta is called the most general unifier.

All variables in the given two literals are implicitly universally quantified.

To make literals match, replace (universally-quantified) variables by terms

The unification routine, UNIFY is to take two atomic sentences p and q and returns α substitution that would make p and q look the same.

UNIFY (p, q) = θ where SUBST ( θ, p) = SUBST (θ, q)

θ = Unifier of two sentences

For example:

p 🡪 S1(x, x)

q 🡪 S1(y, z)

Assume θ = y

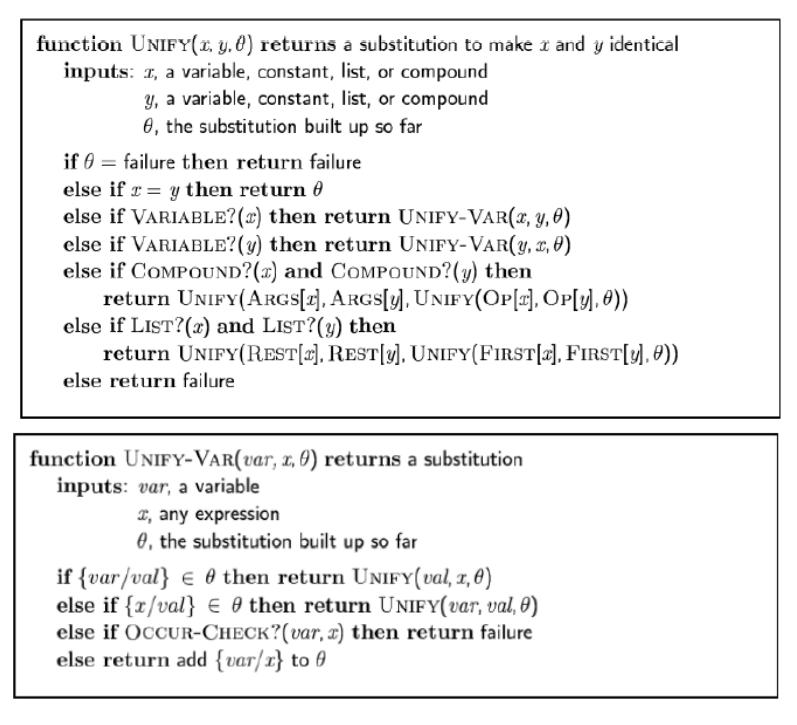
p 🡪 S1(y, y) – x/y (Substituting y for x)

q 🡪 S1(y, y) – z/y (Substituting y for z)

In the above two sentences (p, q) the unifier of two sentences i.e., θ = y is substituted in both the sentences, which derives a same predicatename, same number of arguments and same arguments names.

Therefore the given two sentences are unified sentences.

**Unification Algorithm:**

****