Experiment 06

* Title : Write a program to implement first order logic in python

* Objectives :

1. To understand the basics of first order logic
2. To implement the first order logic

* Theory : First Order logic (FOL) is another way of knowledge representation in AI. It is an extension to Propositional logic (PL). FOL is sufficiently expressive to represent the natural language statements in a concise way FOL is also known as Predicate logic or First-order predicate logic. First order logic is a powerful language that develops information about the objects in a more easy way & can also express the relationship bet^{w} objects FOL does not assume that the world contains facts like PL, but assumes the following:

- Objects : A, B, ~~pe~~ people, numbers, colors, wars, theories, etc.
- Relations : It can be unary relation such as red, round, etc or n-any relation such as sister of, brother of, etc
- Function : Father of, best friend, third innings

As a natural language, FOL has two main parts :

a. Syntax
b. Semantics

Syntax of FOL :

It determines the collection of symbols which is a logical expression in FOL. The basic syntactic elements of FOL are symbols. We write statements in shorthand notation in FOL

Basic elements of FOL :

Constant	1, 2, A, John, ...
Variables	x, y, z, a & b, ...
Predicates	Brother, Father, ...
Function	Sqrt, LeftLegOf, ...
Connectives	∧, ∨, ¬, ⇒, ⇔
Equality	= / ==
Quantifiers	∀, ∃

* Unification :

The unification problem in first order logic can be expressed as follows: Given two terms containing some variables, find, if it exists, the simplest substitution (i.e an assignment of some term to every variable) which makes the two terms equal. The resulting substitution is called most general unifier. The basic unification algorithm is simple. However, it must be implemented with care to ensure that the results are correct.

Begin by making sure that the two expressions have no variables in common. If there are common variables, substitute a new variable in one of the expressions. (Since variables are universally quantified, another variable can be substituted without changing the meaning) Imagine moving a pointer ~~to~~ left-to-right across both expressions until parts are encountered that are not the same in both expressions. If one is a variable, & the other is a term not containing that variable,

1. Substitute the term for the variable in both expression
2. Substitute the term for the variable in the existing substitution set [This is necessary so that the substitution set will be simultaneous]
3. Add the substitution to substitution set.

Application : Unificaton in logic programming

The concept of unification is one of the main ideas behind logic programming, best known through the language Prolog. It represents the mechanism of binding the contents of variables and can be viewed as a kind of one-time assignment. In this operation is denoted by the equality symbol (=), but is also done when instantiating variables. It is also used in other languages by the use of the equality symbol (=), but also in conjunction with many operations including +, -, *, /. Type inference algorithm are typically based on unification

1. A variable which is uninstantiated i.e no previous unifications were performed on it can be unified with an atom, a term, or another uninstantiated variable, thus effectively becoming its alias. In many modern Prolog dialects & in first-order logic, a variable cannot be unified with a term that contains it ; this is also called occurs check.

2. Two atoms can only be unified if they are identical

3. Similarly, a term can be unified with another term if the top function symbol & arities of the terms are identical & if the parameters can be unified simultaneously. Note that this is recursive behaviour

* Conclusion : Hence we have successfully implemented & studied first order logic in AI with knowledge base