Indian Institute of Engineering Science & Technology, Shibpur,

Department of Computer Science & Technology.

8th Semester Artificial Intelligence Laboratory.

ASSIGNMENT- 5

Logic

Duration- 3 periods. **Full Marks (including Viva Voce)-10**

1. **Truth tables for logical expressions (1).**
   Define predicates and/2, or/2, nand/2, nor/2, xor/2, impl/2 and equ/2 (for logical equivalence) which succeed or fail according to the result of their respective operations; e.g. and(A,B) will succeed, if and only if both A and B succeed. Note that A and B can be Prolog goals (not only the constants true and fail).

   A logical expression in two variables can then be written in prefix notation, as in the following example: and(or(A,B),nand(A,B)).

   Now, write a predicate table/3 which prints the truth table of a given logical expression in two variables.

   Example:
   ?- table(A,B,and(A,or(A,B))).
   ```
   true true true
   true fail true
   fail true fail
   fail fail fail
   ```

2. **Truth tables for logical expressions (2).**

   Continue problem P1 by defining and/2, or/2, etc as being operators. This allows to write the logical expression in the more natural way, as in the example: A and (A or not B). Define operator precedence as usual; i.e. as in Java.

   Example:
   ?- table(A,B, A and (A or not B)).
   ```
   true true true
   true fail true
   fail true fail
   fail fail fail
   ```

3. Truth tables for logical expressions (3).

Generalize problem P2 in such a way that the logical expression may contain any number of logical variables. Define table/2 in a way that table(List,Expr) prints the truth table for the expression Expr, which contains the logical variables enumerated in List.

Example:

?- table([A,B,C], A and (B or C) equ A and B or A and C).

```
true true true true
true true fail true
true fail true true
true fail fail true
fail true true true
fail true fail true
fail fail true true
fail fail fail true
```