Ahsanullah University of Science and Technology
Department of Computer Science and Engineering
CSE4108 Artificial Intelligence Lab Fall 2019

Session 1: Basics of Procedural and Declarative Knowledgebase

I. OBJECTIVES

- To be able to use basic elements of Python for procedural programming of knowledgebase;
- To be able to represent query processing environments declaring facts and rules in Prolog.

II. DEMONSTRATION OF USEFUL RESOURCES

Knowledgebase and Queries to a Knowledgebase

A simple knowledgebase (KB) from the Kinship Domain

Object relationships as a KB:

Hasib is a parent of *Rakib*. *Rakib* is a parent of *Sohel*. *Rakib* is a parent of *Rebeka*. *Rashid* is a parent of *Hasib*. If X is a parent of Y and Y is a parent of Z, then X is a grandparent of Z.

List of tuples and sample procedure to manipulate the KB **in Python**:

```
tupleList1=[('parent', 'Hasib', 'Rakib'),('parent', 'Rakib', 'Sohel'),
        ('parent', 'Rakib', 'Rebeka'),('parent', 'Rashid', 'Hasib')]

# Procedure to find the grandchildren of X

X=str(input("Grandparent:"))
print('Grandchildren:', end=' ')
i=0
while(i<=3):
        if ((tupleList1[i][0] == 'parent')&( tupleList1[i][1] == X)):
                for j in range(4):
                        if ((tupleList1[j][0] == 'parent') & ( tupleList1[i][2] == tupleList1[j][1])):
                                print(tupleList1[j][2], end=' ')
        i=i+1
```

Facts and Rules (KB) **in Prolog**:

```
parent('Hasib' , 'Rakib'). parent('Rakib' , 'Sohel'). parent('Rakib' , 'Rebeka').
parent('Rashid' , 'Hasib'). grandparent(X, Z) :- parent(X, Y), parent(Y, Z).

/* [Built-in KB is enhanced with the 4 facts and 1 rule; two 2-place predicates;   3 variables;   full stop
   (.) as the end marker of a clause/ sentence / statement;   :- as ‘if’;  comma (,) as logical AND. ] */

/* Procedure to find the grandchildren of X */

        findGc :- write(' Grandparent: '), read(X), write('Grandchildren: '),
                grandparent(X, Gc), write(Gc), tab(5), fail.
        findGc.
```

***How** can we **modify** the codes to find the grandparents of somebody?*
***Note** that we need to make more changes in Python than in Prolog.*
***Moreover**, we can pose diverse queries to Prolog code and get interpretable answers.*

III. LAB EXERCISE

1) Explore thoroughly the supplementary material provided for this session.
2) Run and analyze the codes demonstrated in this session.
3) Modify the Python and Prolog codes demonstrated above to find the grandparents of somebody.
4) Enrich the KB demonstrated above with ‘brother’, ‘sister’, ‘uncle’ and ‘aunt’ rules in Python and Prolog.