**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Artificial Intelligence (BITS F444/ CS F407)**

**I Semester 2018-19**

**Programming Assignment-5**

**Coding Details**

**(November 15, 2018)**

*Instruction: Type the details precisely and neatly*

1. ID \_\_\_\_\_2017H1030130P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_Santosh Desai\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of Submitted files :
   1. Api.py
   2. Gui.py
   3. Helper.py
   4. Unification.py
   5. Predicateparser.py
   6. Ruleparser.py
   7. Marcus.py
   8. Wumpus.py
   9. Rulefile1.txt
   10. Rulefile2.txt
   11. Predicatefile1.txt
   12. Predicatefile2.txt
   13. Driver.py
2. Total number of submitted files: \_\_\_\_13\_\_\_\_\_\_\_
3. Name of the folder :\_\_\_\_\_\_\_\_2017H1030130P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Have you checked that all the files you are submitting have your name in the top?(yes/no) yes
5. Have you checked that all the files you are submitting are in the folder as specified in 4 (and no subfolder exists)?(yes/no) yes
6. Techniques implemented
   1. Unify algorithm (yes/ no):\_\_\_yes\_\_\_\_\_\_\_\_
   2. Forward Chaining (Yes/No) : \_\_\_\_yes\_\_\_\_\_\_
7. Logic Problems solved manually
   1. Logic Problem 1 (Yes/No) : \_\_\_yes\_\_\_\_\_\_\_

List all queries handled by you:\_\_

* Is Marcus Alive now?
* Who assassinated Caesar?
* Is Marcus loyal to Caesar?
* Who is the Ruler?
* When did the Volcano erupt?
* Is Marcus Dead in 60 AD?
* Did Marcus hate Caesar?
* Is Marcus Alive in 35 AD?
  1. Logic Problem 2 (Yes/No) : \_\_\_yes\_\_\_\_\_\_\_

List all queries handled by you: These queries are inside the system and not displayed as problem 1.

* Is x neighbor of y
* Is x dangerous?
* Does x exude stench?
* Does x feel breezy?
* Does x Glitter?
* Was x visited?

Are you able to show the path found by logical reasoning of the agent? (yes/no) : \_\_\_\_yes\_\_\_\_\_

1. Data structures used
   1. For defining predicate: A list where first argument is a function name and rest are the function parameters.
   2. Rules: Rules are first converted into cnf form of definite clause while parsing so that each rule has one term in the consequent and a conjunction of terms in the antecedent. They are stored as list of terms.
   3. Knowledge base: It contains collection of strings which are broken down into rules and combined as needed.
2. Implementation Details for Forward Chaining technique
   1. Name the functions/ modules implemented by you: Forward chaining is a function in the api module.
   2. Which data structure(s) did you use to implement forward chaining for FOL? Explain.

Set, list and dictionaries were used.

A set to use unique clauses in the knowledge-base and list and dictionaries to support substitution of values.

* 1. What is the time complexity of the technique? Each query has individual time complexity of polynomial time in number of known facts and rules [without memoization] but overall, it runs in linear time w.r.t KB.
  2. What is the space complexity of the technique? Polynomial space complexity.

1. Implementation Details for Unify() algorithm
   1. How have you tagged variables, constants, lists and compound statements: The variables must begin with lowercase, constants can be numbers or strings beginning with Uppercase and appear as arguments of functions, lists are read as string and contain [] in them, Compound statements are parsed using Stmt class as a tree.
   2. Name the functions/modules that you implemented : unify, occur\_check,unify\_var in unification.py module
   3. Which data structure(s) did you use to implement unify algorithm? Explain.
      1. Lists and tuples were used to handle the arguments
      2. Dictionary was used to keep track of the unification process.
   4. What is the time complexity of the algorithm? Linear w.r.t. input symbol base.
   5. What is the space complexity of the algorithm? Linear w.r.t. input symbol base.
2. Files
   1. Submitted the files consisting the predicates description (yes/no):\_\_\_\_\_\_yes\_\_\_\_\_\_\_
   2. Submitted the files consisting the rules in FOL (yes/no):\_\_\_\_\_\_\_\_\_yes\_\_\_\_\_\_\_\_\_
3. Graphics: Created the graphics (yes/no)\_\_\_\_\_\_\_yes\_\_\_\_\_\_\_\_\_\_\_
4. Compilation Details:
   1. Code Compiles (Yes/ No):\_\_\_\_\_yes\_\_\_\_\_\_\_\_\_
   2. Mention the .py files that do not compile:\_\_\_\_\_N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. Any specific function that does not compile:\_\_\_\_\_\_\_\_N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   4. Ensured the compatibility of your code with the specified Python version(yes/no)\_\_\_\_yes\_\_\_\_\_\_\_\_
   5. Instructions for compilation of your files mentioning the multi file compilation process used by you (We may use the replica of these for compiling your files while evaluating your code)
      1. Run Driver.py
      2. If option 2 is selected, please kindly wait for the dfs to complete, it may take around 15-20 minutes with 8 GB RAM.
5. Driver Details: Does it take care of the options specified earlier(yes/no):\_\_\_\_\_yes\_\_\_\_\_\_
6. Execution status (describe in maximum 2 lines)

The code executes, albeit slowly for DFS. There’s a version which runs with A\* algorithm but due to increase in branching factor, it takes comparatively more time than DFS for these two models only.

1. Declaration: I, \_\_\_\_\_\_\_\_\_\_\_Santosh Desai\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (name) declare that I have put my genuine efforts in creating the python code for the given programming assignment and have submitted only the code developed by me. I have not copied any piece of code from any source. If the code is found plagiarized in any form or degree, I understand that a disciplinary action as per the institute rules will be taken against me and I will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

ID\_\_\_\_\_\_\_\_\_2017H1030130P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name:\_\_\_\_\_\_\_Santosh Desai\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_15/11/2018\_\_\_\_\_\_\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Should not exceed three pages