BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Artificial Intelligence (BITS F444/ CS F407)
I Semester 2019-20
Programming Assignment-6
Coding Details
(November 29, 2019)

Instruction: Type the details precisely and neatly

1. ID __2016A7PS0150P____
   Name ___Patel_Parth____

2. Mention the names of Submitted files :
   a. utils.py
   b. algo.py
   c. gui.py
   d. coding details PA6
   e. input1.txt
   f. input2.txt
   g. input3.txt

3. Total number of submitted files: __7____
4. Name of the folder :___2016A7PS0150P_____
5. Have you checked that all the files you are submitting have your name in the top?(yes/no) Yes
6. 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

7. Modules implemented:
   a. Created the Bayesian network? Yes
   b. Created Markov blanket? Yes
   c. Created expression from the inputs read ? Yes
   d. Computed probability ? Yes

8. Data structures used:
   a. To represent the Bayesian network:
      class Bayesian_Network is created. It has the following attributes:
      - var_set: A Python set containing all variables as strings
      - graph: A Python dictionary with key as variable (in the form of a string) and value as a Python list of all variables (again in the form of strings) that the key variable points to in the bayesian network.
      - rev_graph: A Python dictionary with key as a variable (in the form of a string) and value as a Python list of all variables (again in the form of strings) that point to the key variable in the bayesian network.
      - cpt: A Python dictionary storing information present in all conditional probability tables. For example, P(A|~B,C)=0.01 is represented by the following entry of the dictionary: cpt[(‘A’, frozenset([‘~B’,’C’]))] = decimal.Decimal(0.01). Likewise, P(~A) = 0.2 is represented as cpt[(‘~A’, frozenset())] = decimal.Decimal(0.2).
   b. To represent Markov blanket:
      The function compute_markov_blanket() returns a Python set of all variables (represented as strings) present in the markov blanket. Markov blanket of any node/variable is computed using the attributes graph and rev_graph of Bayesian_Network class.

c. To represent the variables:
   A variable is represented as a string. For example, positive literal A is represented as 'A' and the corresponding negative literal is represented as '~A'.

d. To represent the expression for probabilistic query:
   The expression for the probabilistic query is represented using two Python dictionaries: QUERY_VARS and CONDITION_VARS. For example, P(A, ~B|~C,D) is represented as follows:
   - QUERY_VARS['A'] = 1
   - QUERY_VARS['B'] = 0 # As its negative literal
   - CONDITION_VARS['C'] = 0
   - CONDITION_VARS['D'] = 1

9. Implementation Details:
   a. How did you create the CPT reading the data from the file?
      All the CPTs are represented using a single Python dictionary (as described above). While reading data(i.e each entry of each CPT) from the file sequentially, I store two values in the dictionary. For example, P(A|B,~C) = 0.01 is read from the file. Then, cpt[('A', frozenset(['B', '~C']))] = decimal.Decimal(0.01) and cpt[('~A', frozenset(['B', '~C']))] = decimal.Decimal(0.99) are stored in the cpt dictionary.

   b. How did you access the BN to obtain the Markov blanket?
      Markov blanket of any node/variable is computed using the attributes graph and rev_graph of Bayesian_Network class, as these attributes easily provide parents, children and children's parents of the specified variable.

   c. How did you access the CPTs?
      As the cpt is a Python dictionary, to access for example P(A|~B,C), we could use the following expression in Python: cpt[('A', frozenset(['~B', 'C']))].

   d. How did you expand the expression for the conditional dependence on variables?
      P(X|E) was split into two parts → P(X,E) and P(E) using Bayes theorem. Then, **variable elimination** was performed on the Bayesian network using X and E as mentioned in section 14.4 of the textbook. Then, topological ordering of all the remaining variables was found out using DFS. Let all the remaining variables be represented as Z={Z1, Z2, …, Zn} in topological order. Then, X∪E∪Y = Z, where Y is the set of all non-eliminated variables that are neither in query nor in evidence/conditional. To compute P(X,E), the expression was marginalized over Y using the topological ordering and the fact that P(Z1, Z2, …, Zn) = **Π**P(Zi|Parents(Zi)). Similarly, we can compute P(E) using marginalisation over X∪Y. Recursion is used to implement marginalisation.

   e. How did you marginalize the expression?
      As explained above, the expression P(X,E) was marginalised over Y using topological ordering of variables, i.e. P(X,E) = $\boldsymbol{\Sigma}_Y$P(X,E,Y), where the summation is done over variables in topological ordering. Similarly, the expression P(E) was marginalised over X∪Y. Recursion is used to implement marginalisation.

   f. How many terms does a query have? Give example.
      A query consists of two parts: QUERY_VARS and CONDITION_VARS. For example, let A,B,C,D,E,F be all nodes in the network. Let query be P(A,~B|C,~D). Then, the query consists of: QUERY_VARS={'A':1,'B':0} and CONDITION_VARS={'C':1,'D':0}.

g. How have you handled the conditional independence of variables?
   As P(Z1, Z2, …, Zn) = ΠP(Zi|Parents(Zi)), we can find all parents of Zi using rev_graph attribute of the Bayesian_Network class and then, its independent of all other variables/nodes in the network.

10. Graphics: Created the graphics (yes/no)__Yes___

11. Output
    a. Execute your program to answer the following probabilistic queries. Mention the answer obtained by your program. Also compute the Markov blanket of the variable A.
       - P(D, A, L| ¬R, X, P, ¬O) = 0.09974328676359072
       - P(A) = 0.2275876805815709375
       - P(F,R|¬A, ¬P) = 0.12814958359294055741911 90194
       - P(D) = 0.47212254678467130800282 50000
       - P(D|P) = 0.5065278266797952298796
       - P(A|¬Y, ¬C) = 0.048956189735653450186554 27558
       - P(A,D|O, ¬R,P) = 0.2242321037883778368
       - Markov Blanket of A = [D, H, Y, N, L, A, F, G, X, C, B]

12. Compilation Details:
    a. Code Compiles (Yes/ No):__Yes_____
    b. Mention the .py files that do not compile:__N/A___________________________________
    c. Any specific function that does not compile:___N/A________________________________
    d. Ensured the compatibility of your code with the specified Python version(yes/no)___Yes___
    e. 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) On ubuntu terminal, use: python gui.py

13. Driver Details: Does it take care of the options specified earlier(yes/no):__Yes____
14. Execution status (describe in maximum 2 lines):
    All the submitted code works. The GUI is as specified - For both query and condition variables, buttons have been provided to select/remove positive/negative literals without violating any of the mentioned constraints. An interface is also provided to display markov blanket of selected variable. Functionality is provided to display query in conventional manner, display computed probability upto 20 decimal places and select input file.

15. Declaration: I, ___Parth_Patel___ (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__2016A7PS0150P______ Name:__Parth_Patel____

Date: __29/11/2019___

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