CptS350 Symbolic Graph Project (12 Points)

1. Make yourself familiar with Python and pyEDA package (read the example code in the documentation of the package). You will find installation instructions at https://pyeda.readthedocs.io/en/latest/install.html.

2. Look at your class notes on how a graph is represented in a Boolean formula and then how a Boolean formula is represented in BDD, and on how the transitive closure is computed, in particular, looking at the example of computing the transitive closure of $R \circ R$ (R compose R).

3. Let G be a graph over 32 nodes (namely, nodes 0, ⋯, node 31). For all $0 \leq i, j \leq 31$, there is an edge from node i to node j iff $(i + 3)\%32 = j\%32$ or $(i + 8)\%32 = j\%32$. (% is the modular operator in C; e.g., $35\%32 = 3$) A node i is even if i is an even number. A node i is prime if i is a prime number. In particular, we define [even] as the set {0,2,4,6,⋯,30} and [prime] as the set {3,5,7,11,13,17,19,23,29,31}. We use R to denote the set of *all edges* in G.

4. (graded on correctness and clarity. If you use explicit graph search such as DFS, you receive 0.) (coding in **Python**) Every finite set can be coded as a BDD. You must write a Python program to decide whether the following is true:

   (**StatementA**) for each node u in [prime], there is a node v in [even] such that u can reach v in even number of steps.

Your code shall implement the following steps.

   4.1. Obtain BDDs *RR, EVEN, PRIME* for the finite sets R, [even], [prime], respectively. Pay attention to the use of BDD variables in your BDDs.

   4.2. Compute BDD *RR*2 for the set $R \circ R$, from BDD *RR*. Herein, *RR*2 encodes the set of node pairs such that one can reach the other in two steps.

   4.3. Compute the transitive closure RR2star of RR2. Herein, RR2star encodes the set of all node pairs such that one can reach the other in even number of steps.

   4.4. Compute the BDD PE, from BDDs PRIME, *EVEN*, and *RR*2*star*, that is to encode the set of all node pairs (u,v) such that u is prime and v is even and u can reach v in even number of steps.

   4.5. (**Bonus: This part will add up to 5 pts to your existing midterm score if successfully completed**) Here comes the most difficult part. You need to formulate **StatementA** in terms of BDD operations on the BDD *PE*. There are two quantifiers in **StatementA**: one is "for each", and the other is "there is". First, from what you have learned from Math216 (discrete math), "for each" can be expressed through "there is". Second, "there is" can be implemented using existential quantifier elimination method *BDD.smoothing*(). As a result, the entire **StatementA** is a BDD without free variables and hence it is either true or false. Return the truth value.

Many students find methods *BDD.compose*() and *BDD.smoothing*() are quite useful in the package.

5. You need to turn-in two files for the assignment:
   5.1. working python source code named 'BDDproject.py', and
   5.2. a screenshot of your code execution results

   Please zip your files! Make sure to put comments in your code so it is readable. Your TA will run your code!