Introduction to Software Verification – HW No. 4

Winter 2022-2023

TA in charge: Omer Rappoport

<u>Please note</u> that answers without an explanation will not be checked.

Question 1

Let $G = (V, E)$ be a directed graph and let $w: E \rightarrow \{0,1\}$ be a weight function for its edges. The weight of a path is the sum of its edges' weights.
In addition, let A, B be subsets of V.

The graph is represented by the following BDDs:

- $V(\bar{v})$ - represents the set of vertices.
- $E_0(\bar{v}, \bar{v}')$ - represents the set of edges that weigh 0.
- $E_1(\bar{v}, \bar{v}')$ - represents the set of edges that weigh 1.
- $A(\bar{v})$ - represents the set A of vertices.
- $B(\bar{v})$ - represents the set B of vertices.

a. Construct a BDD $A'(\bar{v})$ that represents all the vertices in A whose neighbors in G are in B (regardless of edge weight).

b. Construct a BDD $V_{2,1}(\bar{v}, \bar{v}')$ that represents all pairs of vertices in G such that there exists a path between them with length 2 and weight 1.

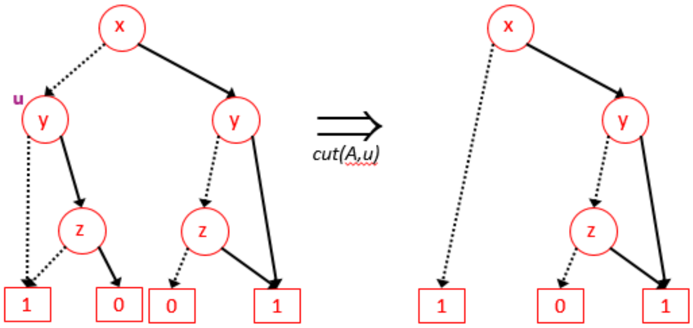
c. Design a BDD-based symbolic algorithm, as efficient as possible, that returns the weight of a shortest path (weight wise) that starts at a vertex in A and ends at a vertex in B. If there is no such path, the algorithm should return -1. Note that there is a path between every vertex and itself with no edges and a weight of 0.
Explain the correctness of the algorithm.

Question 2 - BDD operations

We define a new operation, denoted $cut(A, u)$, where A is a BDD and u is a vertex in A. This operation creates a new BDD A' from A by performing the following actions:

a. Set $low(u) = high(u) = 1$, i.e., the pointers to u's sons are redirected to the leaf 1.
b. Apply the reduce operation to the BDD that was created in the previous step.

For example:



Let $M = (S, R, L)$ be a Kripke structure where S, R are BDDs that represent the set of states and the transition relation, respectively.

Answer the following questions:

a. Let B be a BDD that represents a subset D of S and let u be a vertex in B. We define $B' = cut(B, u)$ and denote by D' the set represented by B'.
   Do the following inclusions hold?
   If yes - explain why. If no - show a counter example.
   1. $D \subseteq D'$
   2. $D' \subseteq D$
b. Let u be a vertex in the BDD R. We define $R' = cut(R, u)$. Consider the Kripke structure $M' = (S, R', L)$.
   1. Do the following inclusions hold?
      If yes - explain why. If no - show a counter example.
      i. $R \subseteq R'$
      ii. $R' \subseteq R$
   2. Is R' necessarily a legal transition relation?
      If yes - explain why. If no - show a counter example.
c. Assume $R' \subseteq S \times S$. Let s be a state in S and let p be an atomic proposition.
   1. Does $M', s \vDash AGp$ necessarily imply $M, s \vDash AGp$? Explain your answer.
   2. Does $M', s \vDash EGp$ necessarily imply $M, s \vDash EGp$? Explain your answer.

Question 3

Let $(S, Tr, Enter, Exit, Princess, Dragon)$ be a maze, where S is the set of squares in the maze, Tr is the set of legal transitions between squares, $Enter, Exit \subseteq S$ are the entrance and exit squares, respectively, and $Princess, Dragon \subseteq S$ are the squares with a princess and a dragon, respectively.

In the maze, there may be one princess or one dragon in every square, but not both (i.e., $Princess \cap Dragon = \emptyset$). Also, it is assumed that there are no cycles in the maze.

A knight traverses the maze. If he reaches a square with a princess, he may take her as bail (but he is not required to do so). If the knight reaches a square with a dragon, he can proceed in his journey only if he has a princess to present to the dragon.

A path in the maze from an entrance square to an exit square is called safe if whenever the knight reaches a square with a dragon, he has a princess with him to present to the dragon.

The graph is represented by the following BDDs:

- $S(\bar{v})$ - represents the set of squares.
- $Tr(\bar{v}, \bar{v}')$ - represents the set of legal transitions.
- $Enter(\bar{v})$ - represents the set of entrance squares.
- $Exit(\bar{v})$ - represents the set of exit squares.
- $Princess(\bar{v})$ - represents the set of squares with a princess.
- $Dragon(\bar{v})$ - represents the set of squares with a dragon.

The knight wants to check whether there is a safe path in the maze to ensure he can traverse it safely.

a. Assume that every path from an entrance square to an exit square has at most one square with a dragon. Help the knight and design a BDD-based symbolic algorithm, as efficient as possible, that computes the set of all entrance squares with safe paths starting from them.

b. (optional) Solve the above question without assuming that every path from an entrance square to an exit square has at most one square with a dragon.