Brenden Nelson
ID#11635716

350 HW 6

(1) Design an efficient algorithm to count the number of paths from U to V.

My Algorithm.

- 1. run topological-sort on G.
- 2. Forward propogate to get number of paths.

(2) Design an efficient algorithm to count the number of "good" paths from U to V.

My Algorithm

- 1. run topological-sort on G.
- 2. Forward propogate to update table w/ yellows, greens, total # of paths
- 3. if #Yellow > #Green, discard path as "bad", else path is "good".

(3) Υ = regular expression on colors. An ugly path is one that the color sequence on path satisfies Υ.

Design an algorithm to count the number of ugly paths from U to V.

My Algorithm

- 1. Enumerate G as an FA (M_1) with initial state U & accepting state V. Enumerate regular expression Υ as an FA (M_2).
- 2. Run Cartesian product on M_1 & M_2 to get one FA, M that shows all G satisfying conditions Υ.
- 3. Run Tarjans algorithm to determine whether G contains any SCC's.
- 4. If G has an SCC, return $+\infty$, else, (go to 5)
- 5. Run topological-sort on G.
- 6. Forward propogate to get number of paths.

⑥ Design an efficient algorithm to compute the number of binary strings with length n that satisfy the regular expression $((0+11+101)^*(1101))^*$.

My Algorithm

- 1. Enumerate three FA's M_1 and M_2 and M_3.
- M_1: Graph G representing binary.
- M_2: binary strings with length n.
- M_3: regular expression $((0+11+101)^*(1101))^*$
- 2. Run Cartesian product on $M_1 \times M_2 \times M_3$ to obtain FA, M.
- 3. Run tarjans algorithm to determine if G has any SCCs.
- 4. If G has an SCC, return ∞. else (go to 5)
- 5. Run topological sort on G
- 6. Forward propagate to get total # of paths.