Software Specifications
Convert State Diagram to Regex

Cain Susko
February 1, 2022

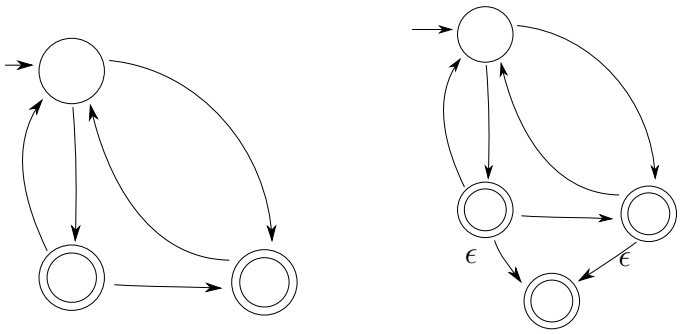
Queen's University
School of Computing

State Elimination Algorithm

This algorithm is used to covert a State Diagram to it's corresponding Regex. The conditions for the algorithm are as follows:

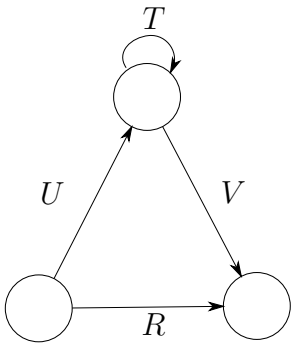
1. The State Diagram used as input should have exactly one final state which is not the start state

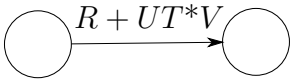
2. If these conditions are not met, we modify the state diagram by adding a new final state with ϵ–transitions from the original final states



The Algorithm uses an intermediate stage *Generalized State Diagrams* where transitions are labelled by Regex. The algorithm eliminates states one at a time. The start state and final state are *not* eliminated.

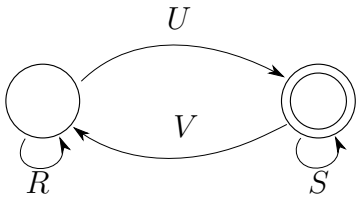
Thus the elimination step is as such, where U, V, T, R are Regex.
We eliminate the top node, the transition between the bottom 2 is now the combined Regex for the removed transitions.





Note, when eliminating a state we need to add transitions to **all pairs** of states connected to the eliminated state. In particular, it is possible that 2 states are the same state.

At the end, the algorithm should produce a 2-state diagram like so:



Which can be translated into it's equivalent Regex.

$$R^*U(S + VR^*U)^*$$

And thus, this is the state elimination algorithm for converting State Diagrams to their corresponding Regex