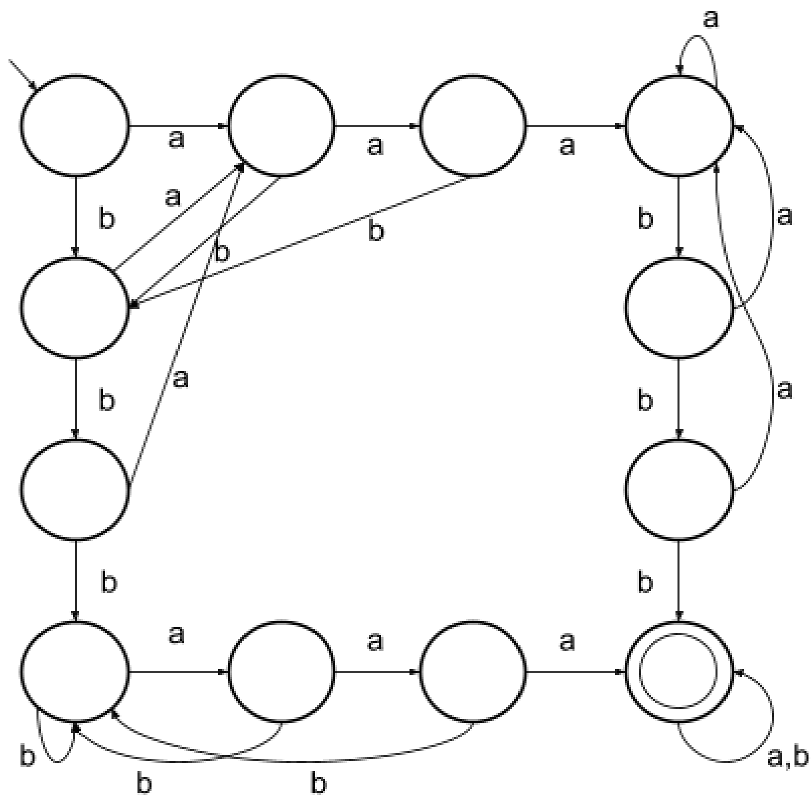
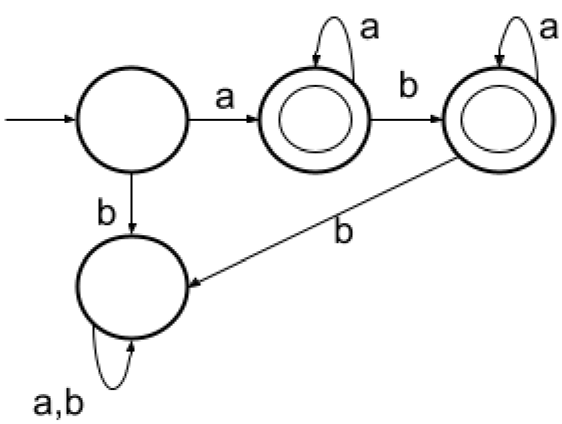
Aughdon Breslin & Isabella Cruz
CS334 PS1 “I pledge my honor that I have abided by the Stevens Honor System.”

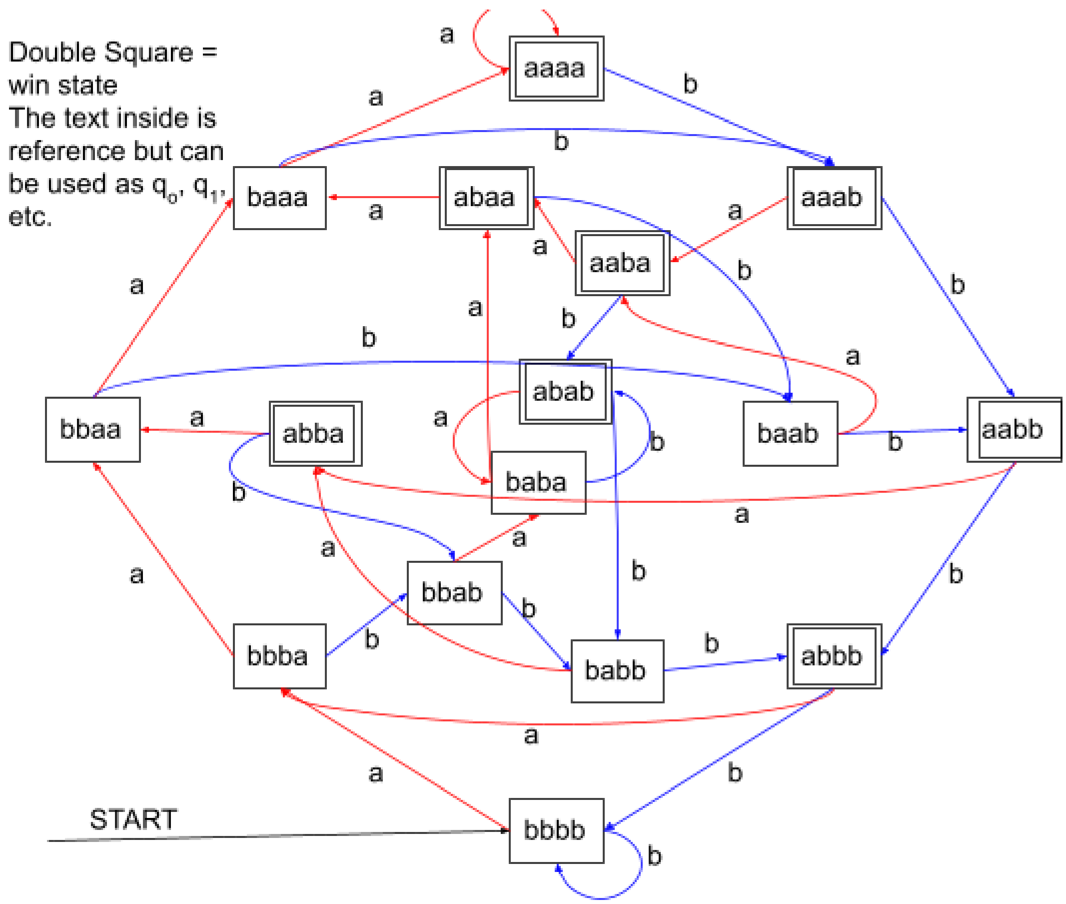
<u>Problem 1</u>



1.



2.



Double Square =
win state
The text inside is
reference but can
be used as q_o, q_1,
etc.
aaaa
baaa
abaa
aaab
aaba
abab
bbaa
abba
baab
aabb
baba
bbab
bbba
babb
abbb
bbbb
START
a
b

3.

Problem 2

PROOF:

Let M_1 recognize A_1 where $M_1 = (Q_1 , \Sigma_1 , \delta_1 , q_1 , F_1)$, and
M_2 recognize A_2 where $M_2 = (Q_2 , \Sigma_2 , \delta_2 , q_2 , F_2)$

Construct M to recognize $A_1 \cup A_2$ where $M = (Q , \Sigma , \delta , q , F)$

Q = $(Q_1 \times Q_2)$ AND ϕ : each state of M is a pair of states, one from M_1 & one from M_2 AND there exists ϕ, which is a dead state

Σ = $\Sigma_1 \cup \Sigma_2$: the alphabet of M includes the alphabets from both M_1 & M_2

δ = $(\delta_1(r_1 , x),\phi) \vee (\phi, \delta_2(r_2 , y)) \vee (\delta_1(r_1 , z), \delta_2(r_2 , z))$

METHOD:

$\delta = Q \times \Sigma \to Q$

$\delta = (Q_1 \times Q_2) \times (\Sigma_1 \cup \Sigma_2) \to (Q_1 \times Q_2)$

$\delta = \{(r_1 , r_2): r_1 \in Q_1 \text{ and } r_2 \in Q_2 \} \times \{x : x \in \Sigma_1 \text{ or } x \in \Sigma_2 \text{ or } x \in \Sigma_1 \cap \Sigma_2 \} \to (Q_1 \times Q_2)$

$\delta = \{ (r_1 , r_2) : r_1 \in Q_1 \text{ and } r_2 \in Q_2 \} \times \{ x : x \in \Sigma_1\}$ OR $\{ (r_1 , r_2) : r_1 \in Q_1 \text{ and } r_2 \in Q_2 \} \times \{ y : y \in \Sigma_2 \}$ OR $\{ (r_1 , r_2) : r_1 \in Q_1 \text{ and } r_2 \in Q_2 \} \times \{ z : z \in \Sigma_1 \cap \Sigma_2 \} \to (Q_1 \times Q_2)$

$\delta(((r_1, r_2), x)) \vee ((r_1, r_2), y)) \vee ((r_1, r_2), z))) = (\delta_1(r_1, x), \phi) \vee (\phi ,\delta_2(r_2, y)) \vee (\delta_1(r_1, z), \delta_2(r_2 , z))$

EXPLANATION:

$(\delta_1(r_1 , x),\phi)$: if a member of the alphabet (x) is only recognized by M_1 and not by M_2 , there exists a transition function δ_1, and clones that would've dealt with δ_2 would die (represented by the empty state ϕ)

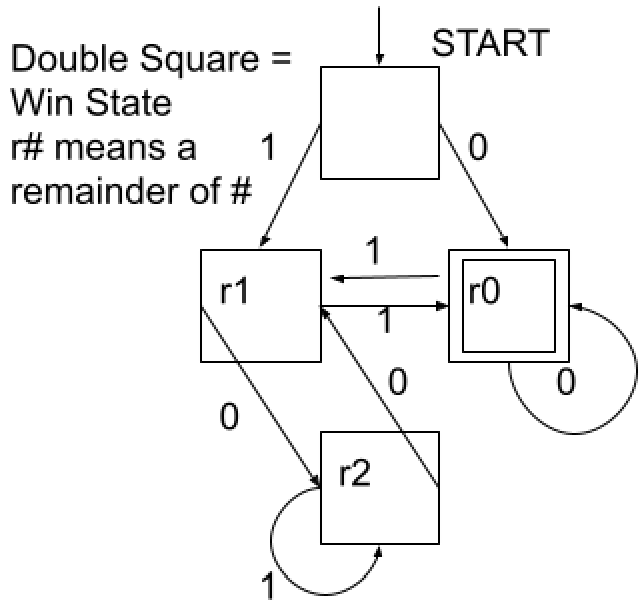
$(\phi, \delta_2(r_2 , y))$: if a member of the alphabet (y) is only recognized by M_2 and not by M_1 , there exists a transition function δ_2, and clones that would've dealt with δ_1 would die (represented by the empty state ϕ)

$(\delta_1(r_1 , z), \delta_2(r_2 , z))$: if a member of the alphabet (z) is recognized by both M_1 and M_2 , there exists both δ_1 AND δ_2

$\mathbf{q_0} = (q_1 , q_2)$: starts M in the start states of M_1 & M_2

F = $(F_1 \times Q_2) \cup (Q_1 \times F_2)$: accept if one of the machines ends in an accept state

Problem 3



1)
2) Prove that Dk is regular, for every $k \geq 1$.
A language is a set of strings.
A language is regular if it is recognized by a finite automaton.
Prove that D_k can be recognized by a finite automaton for every k ≥ 1.

A finite state machine can be made where each state represents a remainder when divided (from a remainder of 0 up to a remainder of n-1 inclusive). This is because our first input can be either a 1 or a 0, which would have a remainder of 1 and 0 respectively for k>1 (for k = 1, both inputs would always result in a remainder of 0). Then any further inputs would do one of two things to the current inputs:

Adding a 1 to the end of the input string would transform the value (n) into 2n + 1. Adding a 0 to the end of the input string would transform this n into 2n.

Any of these new inputs, when divided by k, will fall into one of the remainder categories. Since D_k consists solely of input strings that are divisible by k, those input strings will always end in the state r0 (remainder of 0) and will be accepted by the FSA. Therefore, D_k is regular for every k ≥ 1.