Eric Yang

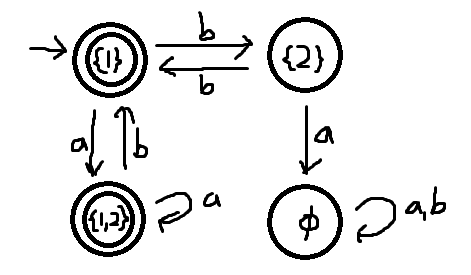
304263623

CS181

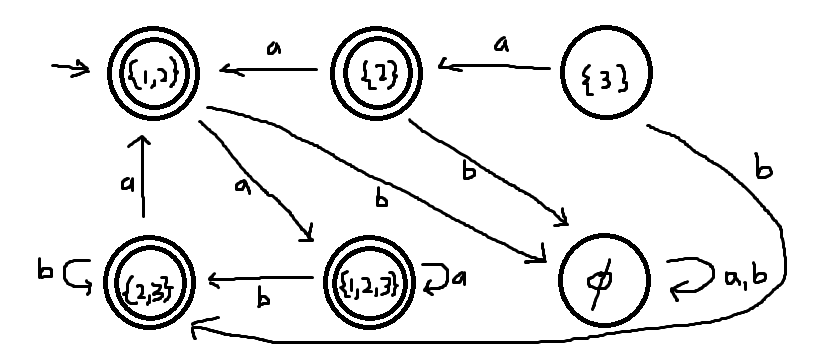
Homework 3

1.16

a.

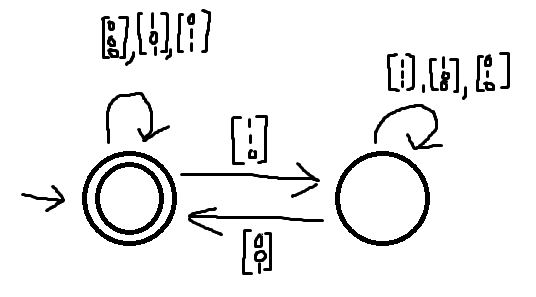


b.



1.32

Show the reverse of B is regular to prove that B is regular. Using addition, an NFA can be created that accepts the reverse of B:



It remains in the accept state if there is no carry. Once there is a carry, it moves to the reject state until either the carry is performed or another carry is generated that causes the NFA to have an error. Since the reverse of B is regular, B is as well.

1.40

a.

Create a DFA for NOPREFIX(A) from the DFA of A. Easier to construct an NFA for complement of NOPREFIX(A). The complement is { w ∈ Σ ∗ : there is a string in A that is a proper prefix of w }.

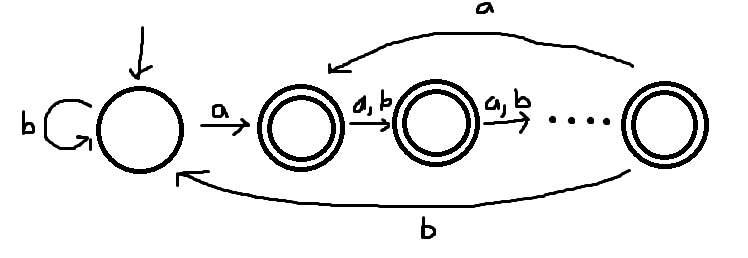
b.

1.41

To construct a DFA for the perfect shuffle of A and B, use the DFAs of A and B. The DFA should constantly switch between the DFAs of A and B after reading every character. If both DFAs are in accept states after the entire string is read, the input is accepted.

1.62

A DFA that could recognize Dk with at most k+1 states can be created and represented with k+1 states. The start state would be the first and only reject state and the accept states would be every other state up to k+1th state. The DFA remains in the start state until it reads an a. Then it starts moving down the chain of states until it either stops on an accept state or loops back to the reject state or first accept state depending if the next character is b or a respectively .



1.66

a.

To create a finite automata M’ that recognizes f(B), it should be possible to swap all the transitions in the DFA of B with their respective homomorphisms. This should provide a DFA for f(B).

1.69

a.

There cannot be a DFA with less than 2^k states because the DFA would need exactly 2^k states to keep track of the possible ways the first w is constructed so that it can be compared to the last half to make sure both w match.

b.