Homework 2

CSC 445-01: Theory of Computation

Matthew Mabrey, Luke Kurlandski

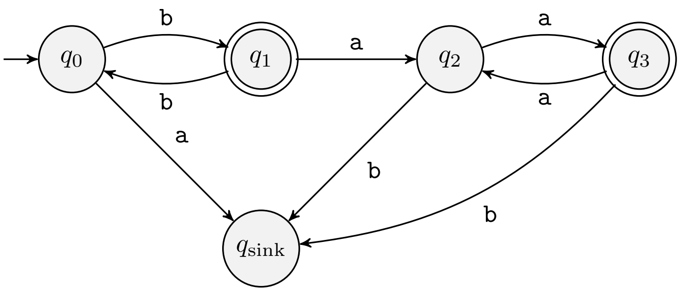
March 5, 2021

1.12

We describe D more simply

$$D = \{w \mid w \text{ is a word with an odd number of } b\text{'s followed by an even number of } a\text{'s}\}$$

And construct a DFA that recognizes D



1.20 g

For the regular expression $R = (\epsilon \cup a)b$

Members include

- b
- ab

Non members include

- a
- aba

1.40 b

Suppose the language A is regular. Then we can construct an NFA N that recognizes A. We can apply a simple modification to N to have it recognize NoExtend(A). Since NoExtend(A) is recognized by an NFA, lets call it N', it is regular. Therefore, the class of regular languages is closed under the NoExtend() operation.

Given the NFA $N = (Q, \Sigma, \delta, q_0, F)$ that recognizes A, we construct $N' = (Q', \Sigma', \delta', q_0', F')$ that recognizes NoExtend(A) where

- $Q' = Q \cup \{q_f\}$: we add one additional state to Q and make it the one and only final state.
- $\Sigma' = \Sigma$: we keep the same alphabet.
- $\delta' = \delta \cup \delta_{\text{new}}$: we keep all of the old transitions but add some new ones. Out of every state, we add a arrow for every letter in Σ that the state did not already have leaving it. This arrow then leads to q_f. This causes the automata to enter the accept state as soon as it reads a character that confirms the string is not a substring of any word accepted by N.

- $q_0' = q_0$: we keep the same start state
- $F' = \{q_f\}$: we make the set of accept states our new state.