COMP 335: Introduction to Theoretical Computer Science

Assignment 3

Nathan Grenier

October 20, 2024
Fall 2024

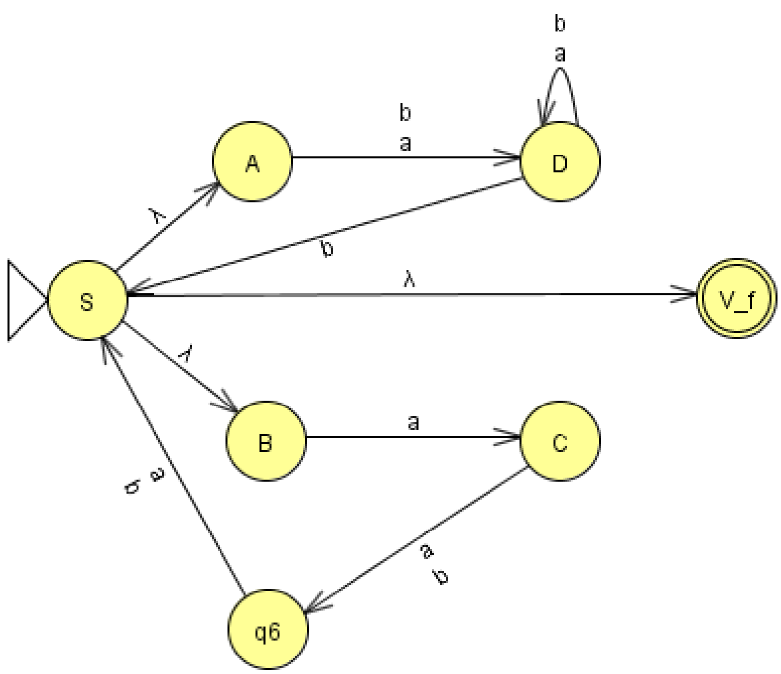
1. [20 Points] For each of the following languages over $\Sigma = \{a, b\}$, write a regular grammar and then convert it into an equivalent NFA using the procedure described in class.

   (a) (10 Points) $L(r)$ where $r = ((a+b)(a+b))^*b + a((a+b)(a+b))^*$

   G:

   - $S \rightarrow A \mid B \mid \lambda$
   - $A \rightarrow aD \mid bD$
   - $D \rightarrow aD \mid bD \mid bS$
   - $B \rightarrow aC$
   - $C \rightarrow aaS \mid abS \mid baS \mid bbS \mid$

   Equivalent NFA of G:

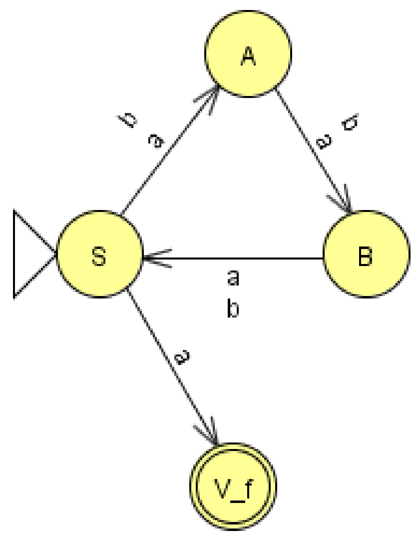


(b) (10 Points) $\{w \in \{a, b\}^* : w$ ends in a and $|w| \equiv 1 \pmod{3}\}$

G:

- $S \rightarrow aA \mid bA \mid a$
- $A \rightarrow aB \mid bB$
- $B \rightarrow aS \mid bS$

Equivalent NFA of G:



2. [25 Points] Fix an alphabet Σ. For any string w with $| w | \geq 2$, let $skip(w)$ be the string obtained by removing the first two symbols of w. Define 2 operators on languages:

$$f_1(L) = \{w \in \Sigma^* : skip(w) \in L\}$$

$$f_2(L) = \{skip(w) \in \Sigma^* : w \in L\}$$

(a) (5 Points) Consider $L' = L(bba^*)$ over tha alphabet $\Sigma = \{a, b\}$. Write a regular expression representing $f_1(L')$. Write another regular expression representing $f_2(L')$.

$$r_1 = (a + b)(a + b)bba^*$$

$$r_2 = a^*$$

(b) (10 Points) Claim: For every regular language L the language $f_1(L)$ is regular. Clearly state whether the claim is TRUE or FALSE, and then prove your answer.

Answer: TRUE

Proof:

(c) (10 Points) Claim: For every regular language L the language $f_2(L)$ is regular. Clearly state whether the claim is TRUE or FALSE, and then prove your answer.

Answer: FALSE

Proof:

3. [20 Points] For each of the following languages, use the Pumping Lemma and/or closure properties of regular languages to show that the language is not regular.

   (a) (10 Points) $L_1 = \{0^k 1^l : k \geq l^4 \geq 0\}$

   (b) (10 Points) $L_2 = \{a^n : n \text{ is not a perfect cube}\}$