Middle East Technical University Department of Computer Engineering

CENG 280

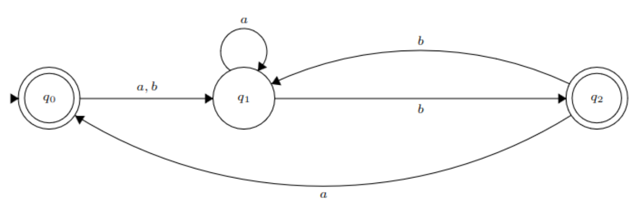
Formal Languages and Abstract Machines

Spring 2023-2024

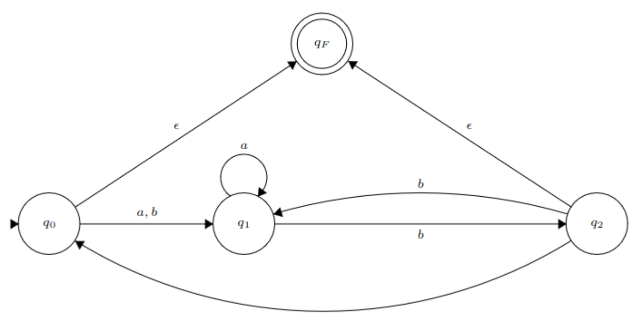
Homework 2 - Solutions

Question 1

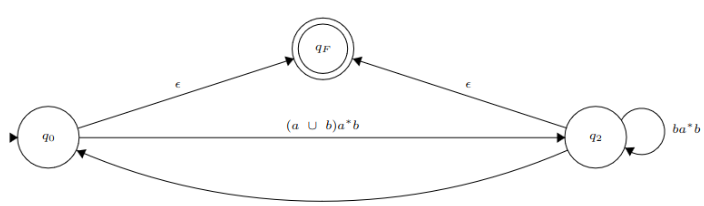
1. Given (D)FA:



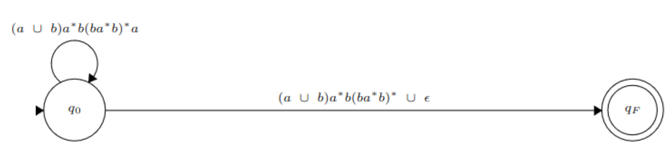
Equivalent NFA with single final state without outgoing transitions:



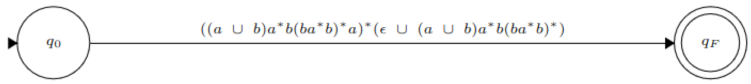
Equivalent GFA 1 (remove q_1):



Equivalent GFA 2 (remove q_2):

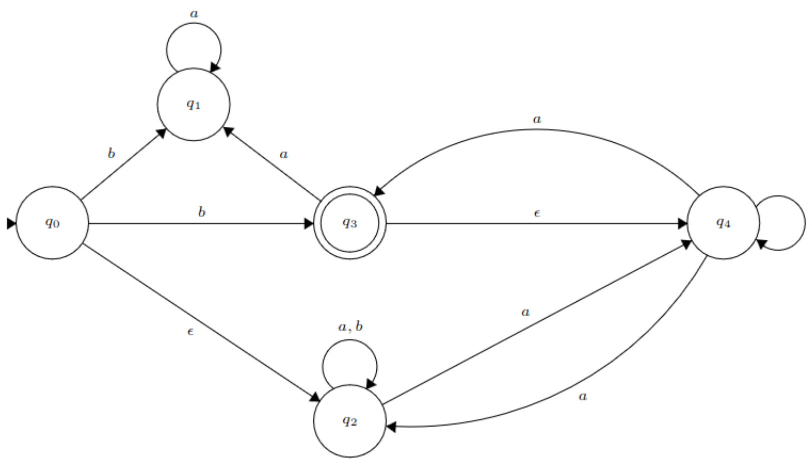


Equivalent GFA 3 (merge transitions):

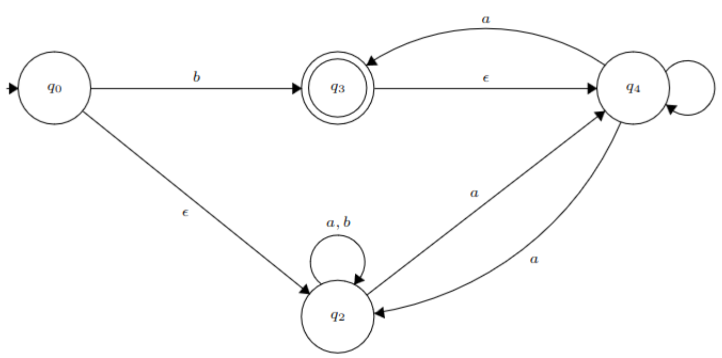


Corresponding regular expression: $((a \cup b)a^*b(ba^*b)^*a)^*(\epsilon \cup (a \cup b)a^*b(ba^*b)^*)$

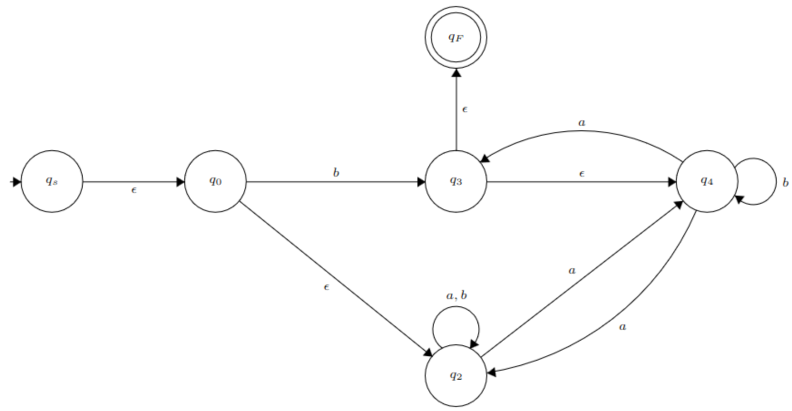
2. Given (N)FA:



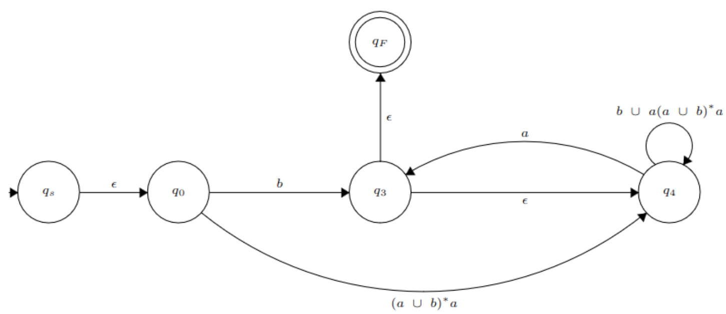
Remove trap states (i.e. q_1):



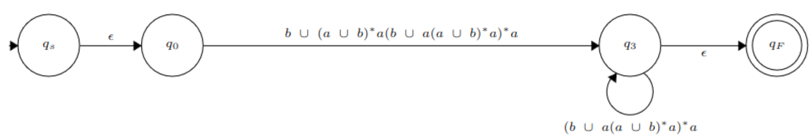
Add dummy initial and final states:



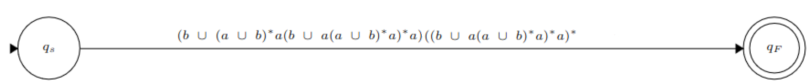
Equivalent GFA 1 (remove q_2):



Equivalent GFA 2 (remove q_4):



Equivalent GFA 3 (remove q_3 and q_0 -trivial-):

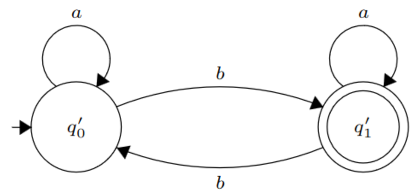


Corresponding regular expression:

$$(b \cup (a \cup b)^*a(b \cup a(a \cup b)^*a)^*a)((b \cup a(a \cup b)^*a)^*a)^*$$

Question 2

1. Where $q'_0 = \{q_0, q_1, q_3, q_4\}$, $q'_1 = \{q_2, q_5\}$,



2. $[\epsilon] = a^* \cup Lba^*$
   $[b] = L$

Question 3

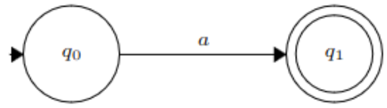
Consider the strings that are in the language. $u = (m + n - k)/2$ given.
There are infinitely many different u values since the $(m + n - k)/2$ can be equal to any natural number when different values are given to m, n and k under defined restrictions (i.e. $m, n, k, u \in \mathbb{N}$). That is, if we divide each the strings in the language as $w_1 \circ w_2$ while $w_1 = a^n b^m c^k$ and $w_2 = d^u$, there are infinitely many w_1 prefices each are required to be appended a different w_2 suffix so as to create a string that is in the language. In other words, there are infinitely many equivalence classes, by the dfinition of the equivalence class.
According to MyHill-Nerode Theorem, a regular language has finitely many equivalence classes. So, the given language is not regular.

Assume that L is regular. Then, the pumping lemma for regular languages applies to L. Consider the string $\omega = a^{n+1}b^n$ which is obviously in the language. By the pumping lemma, w can be rewritten as xyz such that $|xy| \leq n$ where y is nonempty. That is, y can be pumped $i \in \mathbb{N}$ times. More formally, $xy^iz \in L$ must hold for each $i \in \mathbb{N}$. However, when $i = 0$, xy^iz becomes $a^{n-k}b^n$ with $k \geq 0$, which is not in the language. This contradicts the pumping lemma, thus L cannot be regular.

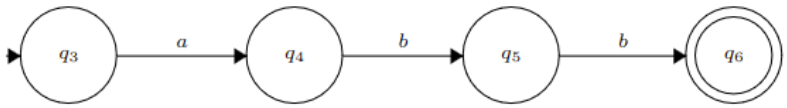
Assume $\overline{L''}$ is regular. Than its complement $\overline{\overline{L''}}$ must also be regular. However, $\overline{\overline{L''}} = L''$ and L'' is not regular. Contradiction. $\overline{L''}$ is not regular.

Question 4

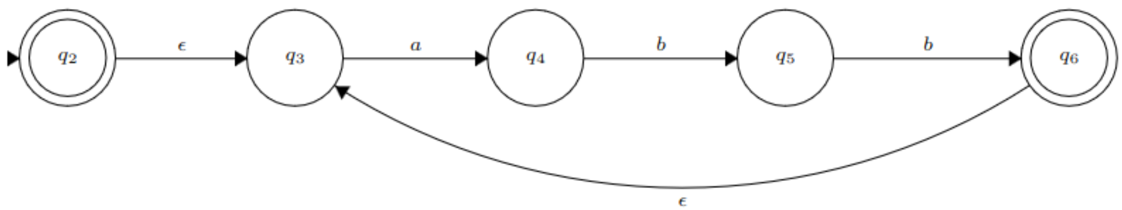
1. NFA for a:



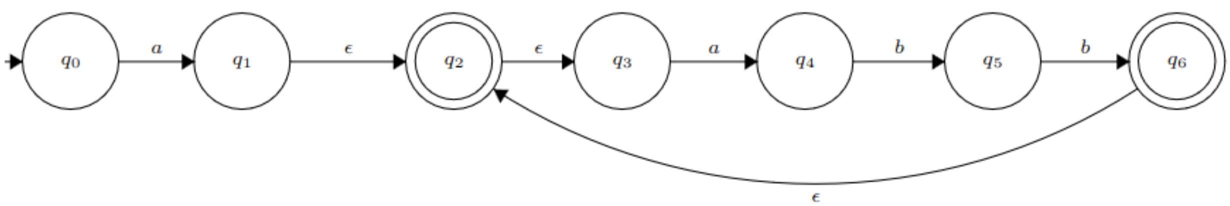
NFA for abb:



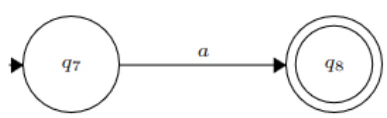
NFA for abb^*:



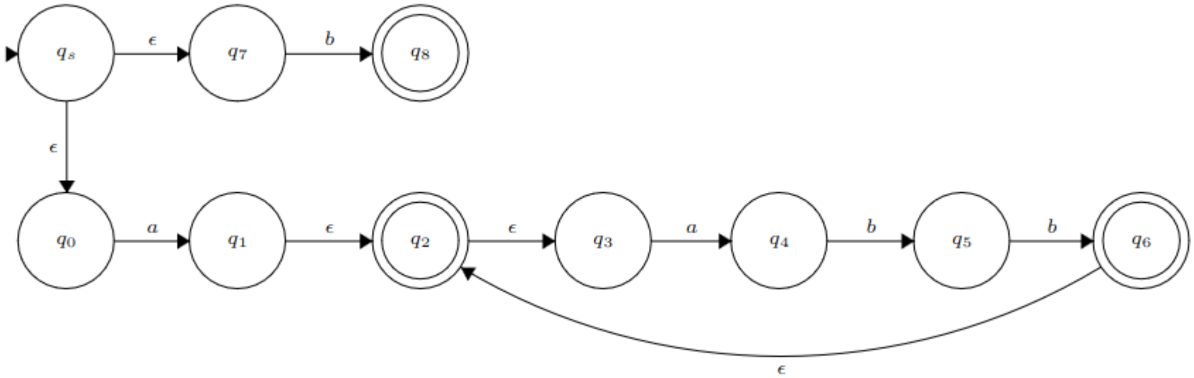
NFA for $a(abb)^*$:



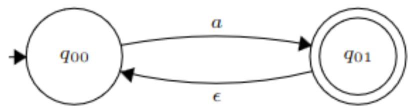
NFA for b:



NFA for $L_1 = L(a(abb)^* \cup b)$:

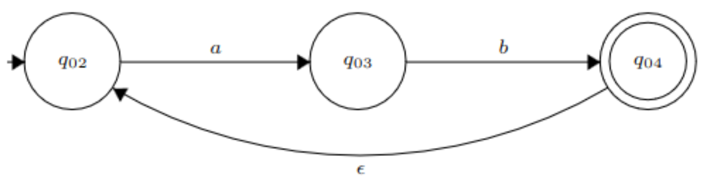


2. NFA for a^+:

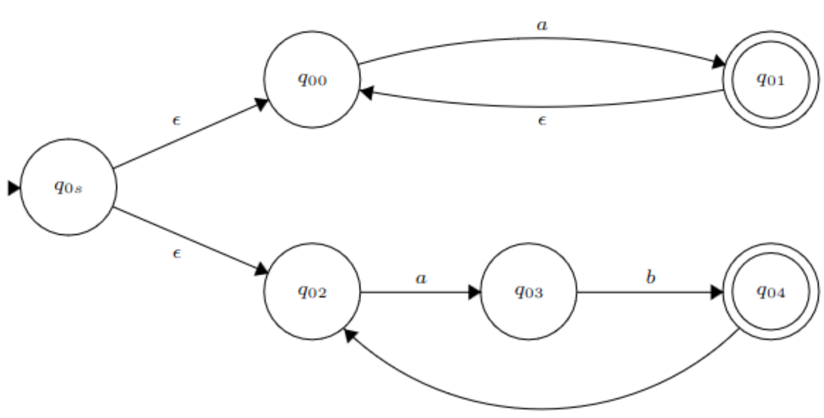


NFA for ab:

NFA for ab^+:



NFA for $L_2 = L(a^+ \cup (ab)^+)$



3. Sketch: Apply standard union construction to the FA recognizing L_1 and the FA recognizing $\overline{L_2}$. What important is, so as to construct the FA recognizing $\overline{L_2}$, you must first convert the NFA constructed at part 2 to a DFA, than swap its final and non-final states (i.e. make final states non-final, and make non-final states final). Note that this complement construction is valid only with DFA but not with NFA.

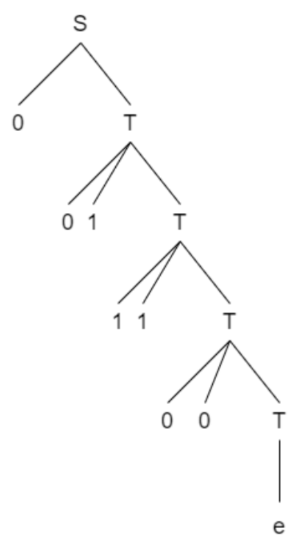
Question 5

1. $G_1 = \{V, \Sigma, R, S\}$ where $\Sigma = \{a, b\}$ $V = \{S, B\} \cup \Sigma$, and $R = \{S \to BbB, B \to BbB, B \to BB|aBb|bBa|b|e\}$

2. $G_2 = \{V, \Sigma, R, S\}$ where $\Sigma = \{0, 1\}$ $V = \{S, A, B\} \cup \Sigma$, and $R = \{S \to AB, A \to 0A1|e, B \to 1B2|e\}$

3. $G_3 = \{V, \Sigma, R, S\}$ where $\Sigma = \{0, 1\}$ $V = \{S, T\} \cup \Sigma$, and $R = \{S \to 0T|1T, T \to 00T|01T|10T|11T|e\}$

   Parse tree for string 0011100:



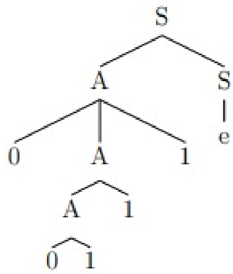
Question 6

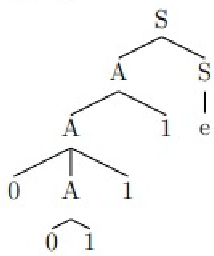
Give the (context-free) languages generated by each of the given grammars:

1. Set of the strings over the alphabet $\{0, 1\}$ that start and end with the same symbol.
2. Set of the strings over the alphabet $\{0, 1\}$ that contain at least two 1's.

Question 7

1. Two different parse trees for the string *00111* can be given as below. Thus, G is ambiguous.





2. Add T to V, and replace R with R' such that $R' = \{S \to T|e, T \Rightarrow AT|A, A \to A1|B, B \to 0B1|01\}$
3. $S \Rightarrow T \Rightarrow A \Rightarrow A1 \Rightarrow B1 \Rightarrow 0B11 \Rightarrow 00111$