Theory of Computation

(CS340)

Closed book exam. Duration: 1 hour

13 September 2010

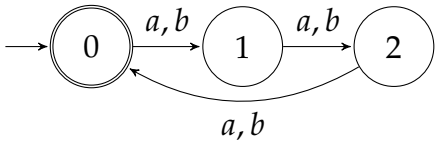
Exercise 1

1. Construct a non-deterministic finite automaton A whose language over $\Sigma = \{a, b\}$ satisfies the following two constraints:
   - All words of $\mathscr{L}(A)$ have a length that is divisible by 3.
   - All words of $\mathscr{L}(A)$ start with the letter a and end with the letter b.

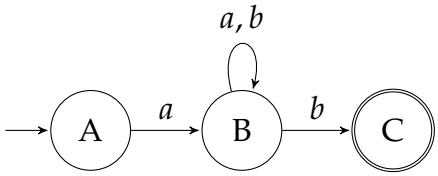
   Please justify your construction.

Answer:

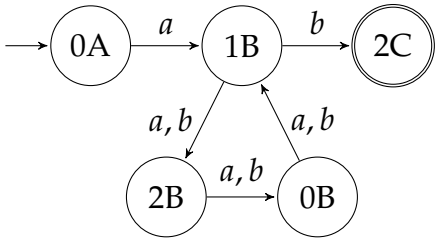
Building a separate automaton for each constraint is easy.
The following automaton recognizes all words $w \in \Sigma^\star$ such that $|w| = 0 \mod 3$.



The following automaton recognizes $a \cdot \Sigma^\star \cdot b$:



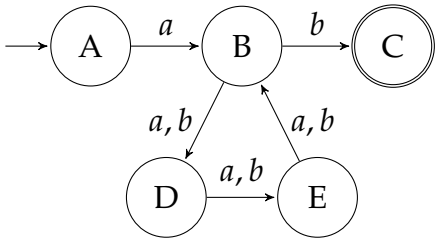
The question was actually to build an automaton that recognizes words that satisfy these **two** constraints, i.e., the intersection of the languages of these two automata. This can be done by building the synchronized product of the two automata:



2. Give a regular grammar that produces $\mathscr{L}(A)$.

Answer:

Since we have finite automaton describing the language, we can just generate a language out of it. Let us just rename the states, so we can use single letters in the grammar.



A regular grammar producing $\mathscr{L}(A)$ can thus be read directly from the automaton:

- $S \to A$
- $A \to aB$
- $B \to bC \mid aD \mid bD$
- $C \to \varepsilon$
- $D \to aE \mid bE$
- $E \to aB \mid bB$

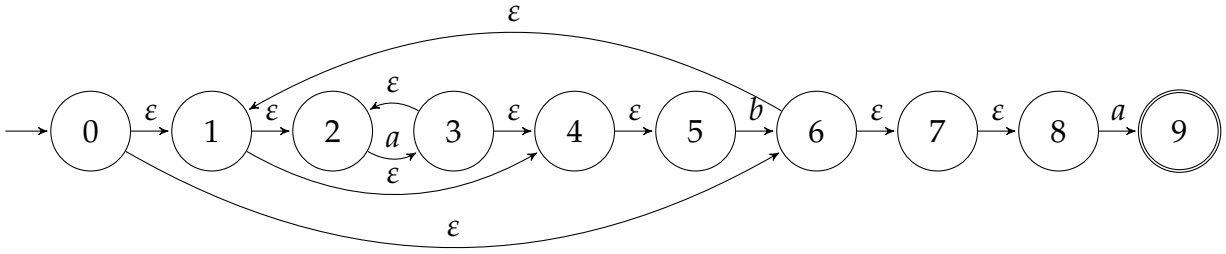
Exercise 2

Consider the regular expression $(a^\star b)^\star a$ defined over $\Sigma = \{a, b\}$.

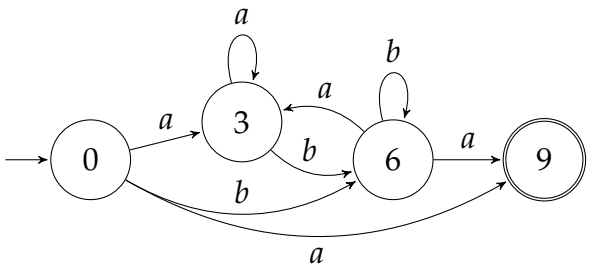
1. Construct a DFA that recognizes $\mathscr{L}((a^\star b)^\star a)$

Answer:

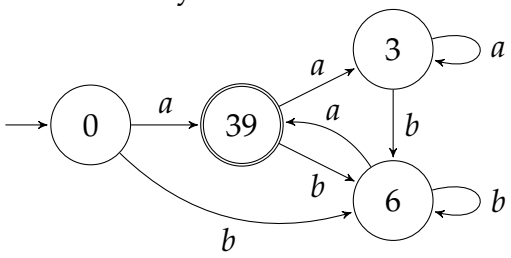
Here is the Thompson automaton for $(a^\star b)^\star a$:



Running the ε-closure on the above automaton yields the following:



But this is only an NFA and we were asked for a DFA, so let's determinize it:

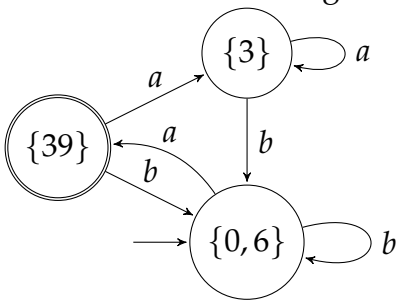


Note that there exist other DFA recognizing the same language. In any case, you should check your automaton to make sure it cannot recognize words that are not in the language (many of you proposed automata that would accept *aa* for instance), and make sure that it does not miss words from the language.

2. Explain why there cannot exist a 2-state DFA that recognizes the same language.

Answer:

If we run the minimization algorithm (e.g. using partition refinement) on the previous DFA, we obtain the following 3-state DFA:

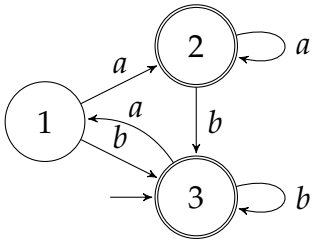


Because this is a minimal automaton we cannot have a DFA with less states (otherwise this automaton would not be minimal).

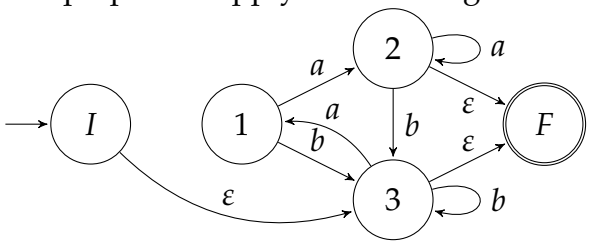
3. Construct a regular expression for the **complement** language. (Please show the steps of Brzozowski and McCluskey's algorithm after each elimination of a state.)

Answer:

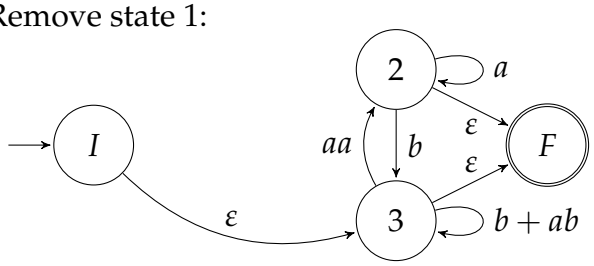
Let us start from the complement of the minimal DFA above:



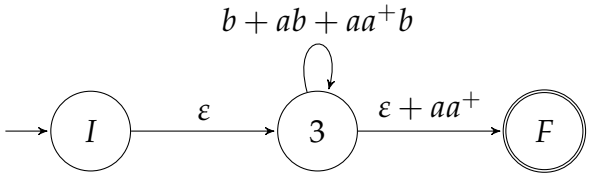
Then prepare to apply the BMC algorithm:



Remove state 1:



Remove state 2:



Finally remove state 3 and the resulting regular expression is $(b + ab + aa^+b)^\star(\varepsilon + aa^+)$. Other equivalent regular expressions are possible if you started from a different automaton or removed the states in a different order.

Exercise 3

1. Prove that the language $L_1 = \{a^n b^m c^n \mid n \in \mathbb{N}, m \in \mathbb{N}\}$ is not regular.

Answer:

That is very easy to prove. We have $L_1 \cap \mathscr{L}(a^\star b^\star) = \{a^n b^n \mid n \in \mathbb{N}\}$. We know that $\mathscr{L}(a^\star b^\star)$ is a regular language (since it is described by a regular expression), and that regular languages are closed under intersection. So if L_1 was regular, so would $\{a^n b^n \mid n \in \mathbb{N}\}$. But we know that $\{a^n b^n\}$ is not regular, so L_1 cannot be.

Some of you tried using the pumping lemma in dubious ways, by considering a words like $a^p b^q c^p = xuy$ without specifying the value of p and q, then considering different cases for u. But the pumping lemma just says that if $a^p b^q c^p$ is large enough you can find a decomposition xuy such that $xu^i y \in L$. With $a^p b^q c^p = xuy$ it could happens that decomposition uses $u \in b^\star$, in which case the word $xu^i y \in L$ really belong to L and you have no contradiction.

A correct way to apply to pumping lemma is as follows. Let K be the constant from the pumping lemma, and consider the word $a^K c^K \in L_1$, then there should exist a decomposition $a^K c^K = xuy$ such that $xu^i y \in L_1$, and you can actually see that not such decomposition is possible by considering the various cases. The trick here is that the pumping lemma should work for any word of L_1 so you should chose one that makes your demonstration easier.

If you want to consider the word $a^K b^2 c^K$ for instance, then you can add that the pumping lemma asserts that we can find a decomposition xuy such that $|xu| \leq K$. This means that $x \in a^\star$, and therefore $xu^i y$ will have too many as.

2. Is is possible to find two **regular** languages L_2 and L_3 such that $L_2 \subseteq L_1 \subseteq L_3$?

Answer:

Yes. $\emptyset$ and $\Sigma^\star$ are regular languages, and obviously $L_2 = \emptyset \subseteq L_1 \subseteq \Sigma^\star = L_3$. This is actually true for any language L_1, not just the one from the previous question.

Exercise 4

Give context-free grammars for the following three languages defined over $\Sigma = \{a, b\}$.

1. $L_4 = \{a^{3i} b^i \mid i \in \mathbb{N}\}$,

Answer:

- $S \to aaaSb \mid \varepsilon$

2. $L_5 = \{a^m b^n \mid \forall m, n \text{ such that } m \geq n \geq 0\}$,

Answer:

- $S \to aSb \mid aS \mid \varepsilon$

3. $L_6 = \{u \in \Sigma^\star \mid \text{the number of } a \text{ in } u \text{ is equal to the number of } b \text{ in } u\}$.

Answer:

- $S \to aSbS \mid bSaS \mid \varepsilon$

Beware that the grammar "$S \to aSb \mid bSa \mid \varepsilon$" would be wrong since it cannot produce $abba$.