Automata Theory - Quiz I (Solutions)

K. Subramani
LCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

1 Problems

1. **Induction:** Show that

$$\sum_{i=1}^{n} i^2 = \frac{n \cdot (n+1) \cdot (2 \cdot n + 1)}{6}.$$

Solution: BASIS: At $n = 1$, the LHS is $\sum_{i=1}^{1} i^2 = 1^2 = 1$. Likewise, the RHS is $\frac{1 \cdot 2 \cdot (2 \cdot 1 + 1)}{6} = 1$. Since the LHS and RHS are equal, the basis is proven.

INDUCTIVE STEP: Assume that for some $k \geq 1$,

$$\sum_{i=1}^{k} i^2 = \frac{k \cdot (k+1) \cdot (2 \cdot k + 1)}{6}.$$

Now observe that,

$$\begin{aligned}
\sum_{i=1}^{k+1} i^2 &= \sum_{i=1}^{k} i^2 + (k+1)^2 \\
&= \frac{k \cdot (k+1) \cdot (2 \cdot k + 1)}{6} + (k+1)^2, \text{ using the inductive hypothesis} \\
&= \frac{(k+1)}{6} \cdot [k \cdot (2 \cdot k + 1) + 6 \cdot (k+1)] \\
&= \frac{(k+1)}{6} \cdot [2 \cdot k^2 + k + 6 \cdot k + 6] \\
&= \frac{(k+1)}{6} \cdot [2 \cdot k^2 + 7 \cdot k + 6] \\
&= \frac{(k+1)}{6} \cdot [2 \cdot k^2 + 4 \cdot k + 3 \cdot k + 6] \\
&= \frac{(k+1)}{6} \cdot [2 \cdot k(k+2) + 3 \cdot (k+2)] \\
&= \frac{(k+1) \cdot (k+2) \cdot (2 \cdot k + 3)}{6}
\end{aligned}$$

However, the last step of the derivation corresponds to substituting $n = (k+1)$ in the RHS of the conjecture and thus we have the LHS = RHS at $n = (k+1)$. Applying the principle of mathematical induction, we conclude that

$$\sum_{i=1}^{n} i^2 = \frac{n \cdot (n+1) \cdot (2 \cdot n + 1)}{6}.$$

□

2. **Language Properties:** Let Σ denote an alphabet and let $L \subseteq \Sigma^*$ denote some language. The Kleene closure of L, viz., L^*, is also a language over Σ. Recall that L^* is defined as:

$$L^* = \cup_{i=0}^{\infty} L^i.$$

Argue that $(L^*)^* = L^*$, for any language L.

Solution: First note that as per the definition of Kleene closure, we must have, $X \subseteq X^*$, for any language X. Accordingy, $L^* \subseteq (L^*)^*$. The non-trivial part lies in showing that the converse is true.

Lemma 1.1 $L^* \cdot L^* = L^*$.

Proof: Clearly, $L^* \subseteq L^* \cdot L^*$, since any string $w \in L^*$ can be written as $w = w \cdot \lambda$. Let $w \in L^* \cdot L^*$. We can express w as $w_1 \cdot w_2$, where $w_1, w_2 \in L^*$. But if w_1 and w_2 are in L^*, then so is their concatenation, as per the definition of Kleene closure! This implies that $w \in L^*$ and hence, $L^* \cdot L^* \subseteq L^*$.

It follows that $L^* \cdot L^* = L^*$. □

Theorem 1.1 $\forall i \geq 1,\ (L^*)^i = L^*$.

Proof: We prove the above theorem inductively.

BASIS: Since, $(L^*)^1 = L^*$, by definition, the basis is proven.

INDUCTIVE STEP: Assume that $(L^*)^k = L^*$, for some $k \geq 1$. Now observe that $(L^*)^{(k+1)} = (L^*)^k \cdot L^*$, as per the definition of the exponentiation operator as applied to languages. Using the inductive hypothesis, we can write that $(L^*)^{(k+1)} = L^* \cdot L^*$. But this immediately implies that $(L^*)^{(k+1)} = L^*$, as per Lemma (1.1). Using the principle of mathematical induction, we can conclude that $\forall i \geq 1,\ (L^*)^i = L^*$. □

As per the definition of Kleene closure,

$$\begin{aligned} (L^*)^* &= \cup_{i=0}^{\infty} (L^*)^i \\ &= \cup_{i=0}^{\infty} L^*, \text{ as per Theorem (1.1)} \\ &= L^*. \end{aligned}$$

□

3. **Regular Expression conversion:** Write a regular expression for the language represented by the FSA in Figure (1).

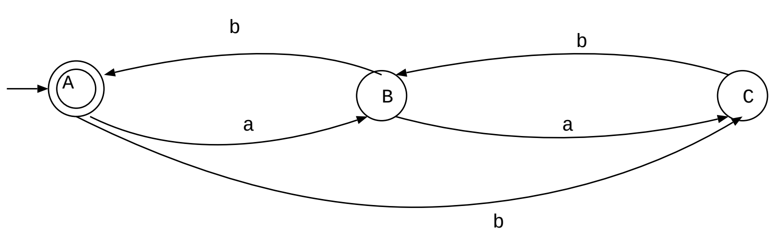

Figure 1: FSA to Regular Expression

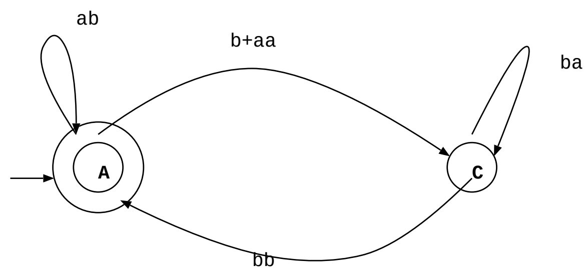

Figure 2: FSA to Regular Expression

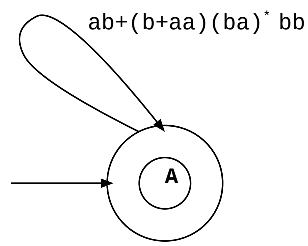

Figure 3: FSA to Regular Expression

Solution: As described in class, we eliminate State B to get the two-state Generalized Transition Graph described by Figure (2).

Eliminating state C, we get Figure (3)

Using R to represent the expression $(ab + (b + aa) \cdot (ba)^* \cdot bb)$, we conclude that the regular expression for the language represented by the input FSA is R^*. $\Box$

4. **Regular Grammars:** Construct a right-linear grammar for the language $L((aab^*)^*)$.

   Solution:

   We first construct the NFA for the given language; it is not hard to see that Figure (4) represents $L((aab^*)^*)$.

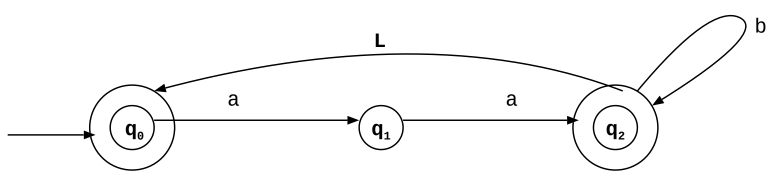

Figure 4: FSA to Regular Grammar

I would like to emphasize the following points:

(i) The L in the figure denotes λ.

(ii) Technically, you should provide an inductive proof that establishes the equivalence between the language $L((aab^*)^*)$ and the NFA in Figure (4). While that is not feasible within the framework of a quiz, you need to convince yourself that the NFA is correct. I did work out the inductive proof though!

As per the algorithm described in class, we can construct the following right linear grammar $G = \langle V, T, S, P \rangle$, where,

(a) $V = \{q_0, q_1, q_2\}$,

(b) $T = \{a, b\}$,

(c) $S = q_0$, and

(d) The productions P are given by:

$$\begin{array}{lcl} q_0 & \rightarrow & a\ q_1 \mid \lambda \\ q_1 & \rightarrow & a\ q_2 \\ q_2 & \rightarrow & b\ q_2 \mid q_0 \mid \lambda \end{array}$$

□

5. **State Minimization:** Minimize the number of states in the DFA shown in Figure (5).

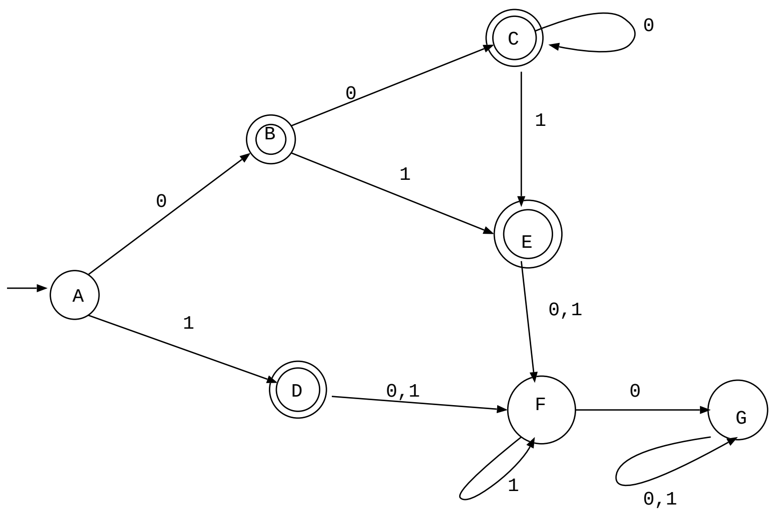

Figure 5: DFA State minimization

Solution: Without loss of generality, we can eliminate one of the two trap states F and G; we choose to eliminate G to get the DFA in Figure (6).

We make the following observations:

(a) States A and F are distinguishable from states B, C, D and E on input λ.

(b) States A and F are distinguishable on 0.

(c) State C is distinguishable from states E and D on input 1; so is state B.

(d) There are no more distinguishable pairs; accordingly, the equivalence classes are: $\{A\}$, $\{B, C\}$, $\{D, E\}$ and F.

Figure (7) represents the minimized DFA.

□

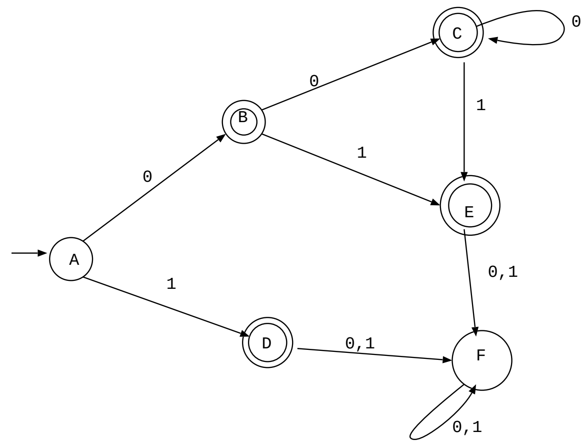

Figure 6: DFA State minimization

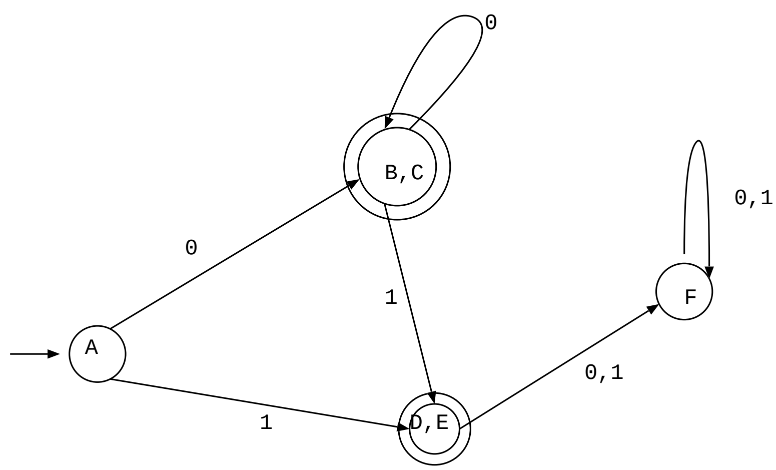

Figure 7: DFA State minimization