Automata Theory - Midterm (Solutions)

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

1 Problems

1. Professor Chikovski wants to prove the conjecture, "If B then C". After working for four hours, he succeeds in proving the theorem, "If A then B and C". His graduate student points out to him that the theorem, "If B then A" is a well known fact. Can the Professor now claim that his conjecture,"If B then C" is a theorem? If so, provide a proof of the same. If not, provide a counterexample.

   Solution: We first write the problem in implication form. Accordingly, the hypotheses of the argument are: (a) $A \rightarrow (B \wedge C)$, and (b) $B \rightarrow A$. From these hypotheses, we wish to conclude that $B \rightarrow C$.

   Formally, we wish to show that Conjecture (1) is a theorem, i.e., it is always **true**.

$$[[A \rightarrow (B \wedge C)] \wedge [B \rightarrow A]] \rightarrow [B \rightarrow C] \tag{1}$$

   The truth-table technique, while correct is too time consuming. Instead, we reason as follows:

$$\begin{array}{rc} & [[A \rightarrow (B \wedge C)] \wedge [B \rightarrow A]] \text{ (hypothesis)} \\ \Rightarrow & [B \rightarrow A] \wedge [[A \rightarrow (B \wedge C)] \text{ (Commutativity of } \wedge) \\ \Rightarrow & B \rightarrow (B \wedge C) \text{ (Properties of Implication; See Homework I)} \\ \Rightarrow & (B \rightarrow B) \wedge (B \rightarrow C) \text{ (Properties of Implication)} \\ \Rightarrow & B \rightarrow C \text{ (Properties of Conjunction)} \end{array}$$

   Since we are able to logically deduce $(B \rightarrow C)$ from the hypothesis, the Professor can indeed claim that his conjecture is a theorem. □

   An approach which is even more intuitive is as follows: Let B be **false**. In this case, the consequence of Conjecture (1), viz., $B \rightarrow C$ is always **true**. The conjunct $A \rightarrow (B \wedge C)$ is $A \rightarrow \textbf{false}$ and the conjunct $B \rightarrow A$ is $\textbf{false} \rightarrow A$. In other words, the hypothesis of Conjecture (1) is $A \rightarrow \textbf{false} \wedge \textbf{false} \rightarrow A$, which is just $A \rightarrow A$ and hence always **true**. Thus, when B is **false**, Conjecture (1) is saying that $\mathbf{true} \rightarrow \mathbf{true}$, which is **true**.

   Now, let B be **true**. In this case, the consequence of Conjecture (1) is C, whereas the hypothesis is $(A \rightarrow C) \wedge A$. But from $A \wedge (A \rightarrow C)$, we can deduce C. Thus, when B is **true**, Conjecture (1) is asking whether $C \rightarrow C$ is **true**. Since this is trivially **true**, the conjecture holds in this case as well.

   Thus Conjecture (1) holds regardless of the value of B, and hence we can conclude that it is always **true**, i.e., "If B then C" is a theorem.

2. Formally prove that the DFA described by the transition table below, accepts all and only those binary strings hichthat do not contain two consecutive $0's$.

	0	1
$\rightarrow * q_0$	q_1	q_0
$* q_1$	q_2	q_0
q_2	q_2	q_2

Proof: We begin by drawing the transition diagram, corresponding to the above transition table.

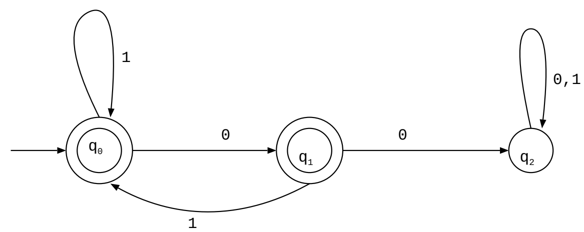

Figure 1: Transition diagram of DFA in Table 1.

Observe that $\Sigma = \{0, 1\}$ for this DFA. We need a few auxiliary results, before stating and proving the main result.

Claim 1.1 *Let w be some string in Σ^*. If w contains a 0, then the first 0 in w causes the automaton to move to state q_1.*

Proof: First observe that $\hat{\delta}(q_0, 1^*) = q_0$, i.e., if the automaton has not seen a 0, it stays in state q_0. We use induction on the position of the first 0 in w. If this 0 is the first character (position 1) of w, then clearly, as per the transition diagram, on seeing this character, the automaton will move to q_1. Let the first 0 occur in position i of w. Then the first $(i-1)$ characters of w are all $1's$. Hence $\hat{\delta}(q_0, w_1 w_2 \ldots w_{i-1}) = \hat{\delta}(q_0, 1^{i-1}) = q_0$, as per the above discussion. Once again, the transition table establishes that on seeing the first 0, the automaton must move to state q_1. $\Box$

Claim 1.2 *Let $w \in \Sigma^*$ be an arbitrary string. If every 0 in w is followed by a 1, then $\hat{\delta}(q_0, w) = q_0$.*

Proof: From Claim 1.1, we know that when the first 0 is seen, the automaton moves to state q_1. If the next character is a 1, then as per the transition table, we are back in state q_0 and the claim is inductively proven. $\Box$

Claim 1.3 *Let $x \in \Sigma^*$ be an arbitrary string. If every 0 in x is followed by a 1, then $\hat{\delta}(q_0, x0) = q_1$.*

Proof: Follows immediately from Claim 1.2 and the transition diagram. $\Box$

Claim 1.4 *If $\hat{\delta}(q_0, w) = q_2$, then $\hat{\delta}(q_0, wx) = q_2$, for all $x \in \{0, 1\}^*$.*

Proof: This is clear from the transition diagram, since we have $\delta(q_2, 0) = \delta(q_2, 1) = q_2$. In other words, once the automaton reaches q_2, it stays in q_2. Therefore, q_2 is a dead state and wx cannot be accepted by the automaton. $\Box$

Note that the automaton has two final states, viz., q_0 and q_1 and one non-final state, viz., q_2.

As per the above discussion, we are required to prove that:

Conjecture 1.1

$$\hat{\delta}(q_0, w) = q_2 \Leftrightarrow \text{w contains two consecutive } 0'\text{s}.$$

Proof:

Only If: We need to show that

$$\hat{\delta}(q_0, w) = q_2 \Rightarrow \text{w contains two consecutive } 0'\text{s}.$$

Observe that if $\hat{\delta}(q_0, w) = q_1$, then $w = x0$ and $\delta(q_0, x) = q_0$. This is because, as per the transition diagram, there is exactly one way for the automaton to get to state q_1 and that is on a 0, from state q_0.

Now, if $\hat{\delta}(q_0, w) = q_2$, then as per the transition diagram, we must have $w = xy$, where, $\hat{\delta}(q_0, x) = q_1$ and $\hat{\delta}(q_1, y) = q_2$. But if $\hat{\delta}(q_0, x) = q_1$, then x must be of the form $u0$, where $\hat{\delta}(q_0, u) = q_0$, as per the above discussion. Likewise, if $\hat{\delta}(q_1, y) = q_2$, then $y = 0v$, as per the transition diagram. We thus see that the last character of x and the first character of y must be $0's$, which establishes that w has two consecutive $0's$.

If: We need to show that:

$$\text{w contains two consecutive } 0'\text{s} \Rightarrow \hat{\delta}(q_0, w) = q_2$$

Let w contain a pair of consecutive $0's$; without loss of generality, we focus on the first occurrence of such a pair and use induction on the position of the first 0 in this pair.

If this first 0 occurs as the first character of w, then $w = 00x$, for some $x \in \Sigma^*$. From the transition diagram, it is clear that $\hat{\delta}(q_0, w) = \hat{\delta}(q_0, 00x) = q_2$ and the claim is proved.

Assume that the claim holds when the first 0, occurs in any position i, where $i \leq n$. Now consider the case, in which the first 0 occurs in position $(n + 1)$ of w. It is clear that in the string $w_1 w_2 \cdot w_n$, every 0 is immediately followed by a 1. As per Claim 1.2, $\hat{\delta}(q_0, w_1 w_2 \ldots w_n) = q_0$. From the transition diagram, the first 0 takes the automaton to state q_1 and the 0, immediately succeeding it, takes the automaton to state q_2, which establishes the claim. □

□

3. Suppose that you are given a DFA $A = (Q, \Sigma, \delta, q_0, F)$, which accepts the language $L \subseteq \Sigma^*$. Let us say that we wish to design a DFA that accepts the language L^c, where $L^c = \{w \mid w \in \Sigma^* \text{ and } w \notin L\}$.

   (i) Argue using induction that the DFA $A^c = (Q, \Sigma, \delta, q_0, Q - F)$ serves the purpose. (2 points)

   (ii) Will the same trick work if A is an NFA. If so, provide a formal proof of the same. If not, provide a counterexample. (3 points)

Solution:

(i) We will show that every string w which leads A to a final state leads A^c to a a non-final state and every string w which leads A to a non-final state, leads A^c to a final state.

Consider a string $w \in L(A)$. Let $|w| = 0$, which means that $w = \epsilon$. Since w is accepted by A, it means that q_0 is a final state of A and hence not a final state of A^c. Therefore w leads A^c to a non-final state. Assume that the claim is true, for all strings of length at most n. Let $w = xa$, where x is a string of length n in Σ^* and a is a symbol in Σ. Since w is accepted by A, $\hat{\delta}(q_0, w) = \hat{\delta}(q_0, xa)$ leads A to final state and hence A^c to a non-final state, which proves the claim.

In identical fashion, we can argue that if $w \notin L(A)$, then $w \in L(A^c)$.

(ii) Consider the NFA in Figure 2, which accepts all binary strings that end in 01.

By complementing the final state, we get the NFA in Figure 3, which not only accepts binary strings that do not end in 01, but also strings that do end in 01. In fact, it accepts $\{0, 1\}^*$!

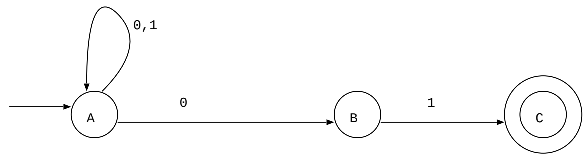

Figure 2: NFA accepting all strings that end in 01.

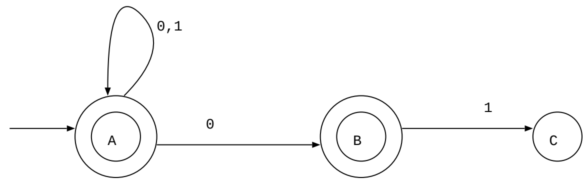

Figure 3: Complement of the NFA

□

4. (a) Convert the regular expression 01^* to a DFA. (2 points)

   (b) Write a regular expression for the DFA described by the following transition table: (3 points)

   Solution:

   (a) Figure 4 is the required DFA.

   (b) We represent the transition table as a transition diagram (Figure 5) and apply the State Elimination technique discussed in class.

   After crushing q_2, we get the DFA, represented by Figure 6

   We then apply the cookie-cutter approach (see Pg. 99 of [HMU01].) As per that approach, $R = (1 + 01)$, $S = 00$, $T = 11$ and $U = 0 + 10$ and the required regular expression is: $(R + SU^*T)^*SU^*$.

□

5. Prove or disprove the following laws on regular expressions:

   (i) $(R + S)^*S = (R^*S^*)^*$. (2 points)

   (ii) $R(S + T) = RS + RT$. (3 points)

   Solution: We use the Concretization theorem and substitute $R = a$, $S = b$ and $T = c$.

   (i) We have to show that

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

   Observe that a is a member of the language $(a^*b^*)^*$, but not a member of $(a + b)^*b$. It follows that the law is incorrect.

	0	1
$\rightarrow q_1$	q_2	q_1
q_2	q_3	q_1
$* q_3$	q_3	q_2

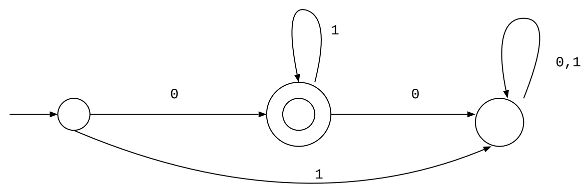

Figure 4: DFA accepting 01^*.

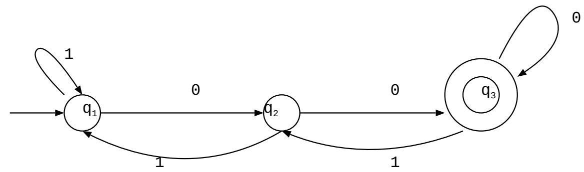

Figure 5: Original DFA

(ii) We are required to show that

$$a(b + c) = ab + ac$$

Observe that both $a(b + c)$ and $ab + ac$ denote precisely the same finite set, viz., $\{ab,\ ac\}$. It therefore follows that the law holds, for all languages R, S and T.

□

6. Let $\Sigma = \{0, 1\}$. Argue that the language $L = \{0^i \cdot 1^j \mid i \geq 0,\ i \leq j\}$ is not regular.

Solution: Let L be regular; therefore $L = L(A)$, for some DFA A. Let n denote the integer of the Pumping Lemma. Since L is an infinite language, I can choose $w = 0^n \cdot 1^{n+1}$ as a member of L. As per the Pumping Lemma, I can break up w into xyz, such that

(i) $y \neq \epsilon$,

(ii) $|xy| \leq n$, and

(iii) $xy^k z \in L, \forall k \geq 0$.

Since $|xy| \leq n$ and $y \neq \epsilon$, y must consist entirely of $0's$; additionally y must contain at least one 0. As per the Pumping Lemma, $xy^{n+2}z \in L$. However, this string clearly contains more $0's$ than $1's$ and hence cannot be in L, as per the definition of L. This is the required contradiction, from which it follows that L cannot be regular. □

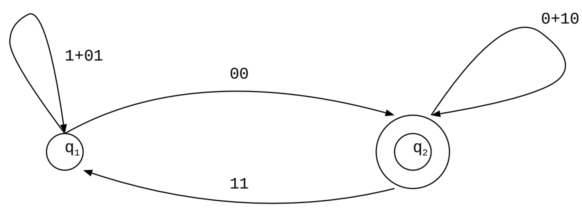

Figure 6: DFA after crushing q_2

References

[HMU01] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *“Introduction to Automata Theory, Language, and Computation”*. Addison–Wesley, 2nd edition edition, 2001.