Homework #2
Automata and Computation Theory
Fall 2018

Written by Eric Rothman

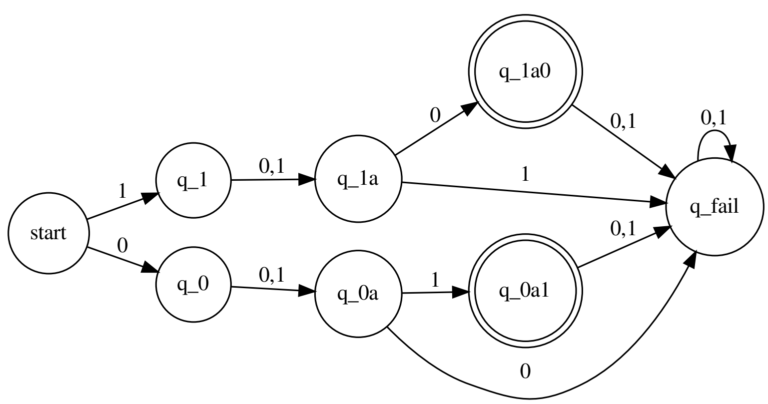
September 27, 2018

1 Problem 1

Give the state diagram of a finite automaton recognizing the following language. The alphabet is $\{0, 1\}$.

$\{$w | w has length exactly 3 and its last symbol is different from its first symbol$\}$

Answer:

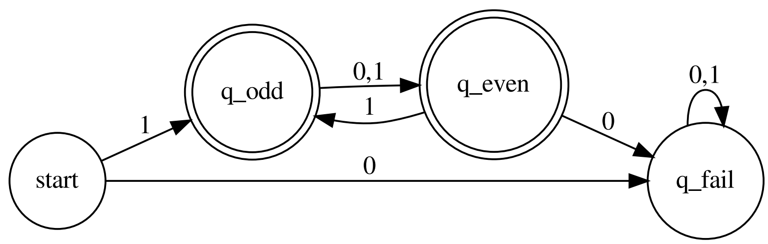


2 Problem 2

Give a finite automaton (both a state diagram and a formal description) recognizing the following language. The alphabet is $\{0, 1\}$.

{w | w is not the empty string and every odd position of w is a 1}

Answer: $M =$



Formal Definiton:

$$M = \{\{start, q\_odd, q\_even, q\_fail\}, \{0, 1\}, \delta, start, \{q\_even, q\_odd\}\}$$

$\delta =$

	0	1
start	q_fail	q_odd
q_odd	q_even	q_even
q_even	q_fail	q_odd
q_fail	q_fail	q_fail

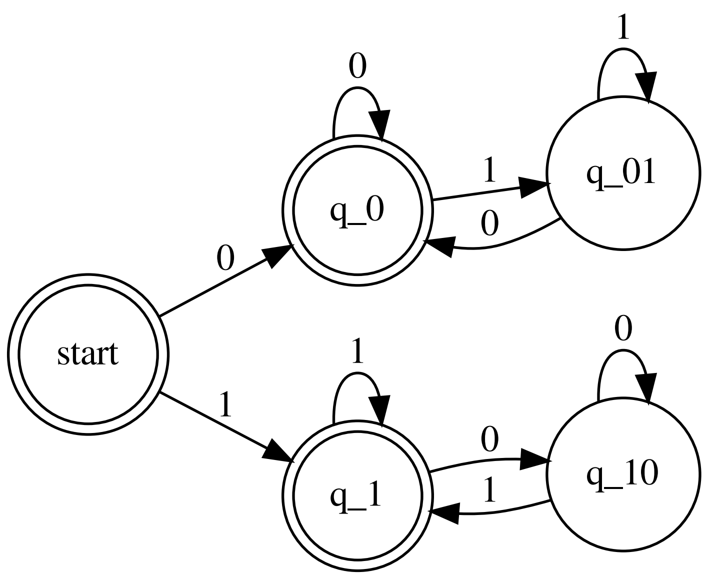
3 Problem 3

Show that the following language is regular, where the alphabet is $\{0, 1\}$.

$\{$w | w contains an equal number of occurrences of the substrings 01 and 10$\}$

Answer:

A simpler way to write out the language is to observe that everytime the string returns to its starting character, there are an equal number of substrings of 01 and 10. I will prove why this is in the proof section of the answer. For a language to be regular, it needs to be the language for a finite automata. I claim and will prove that the given language, A, is the equal to the language, L(M), for the following automata:



Claim 1

I will prove $\{$w | w contains an equal number of occurrences of the substrings 01 and 10$\}$ = $\{$w | w starts and ends on the same symbol$\}$

First Direction

Prove $\{$w | w contains an equal number of occurrences of the substrings 01 and 10$\}$ $\subset$ $\{$w | w starts and ends on the same symbol$\}$

let $a \in A$,

So a has an equal number of occurrences of the substrings 01 and 10

WLOG let a start with 0

Every time the substring 01 appears in a, the current character is a 1.

Since there are an equal number of 01 and 10 substrings in a and since to hava 01 substring, the current symbol must be a 0, the next occurance of one of those two substrings must be 10.

So every time 01 appears, a 10 must appear before another version of 01.

Since there are an equal number of substrings, every 01 substring is followed by a 10 substring. Otherwise there couldn't be an equal amount since either substring needs the other one between iterations.

So the last substring to appear is a 10 substring.

So a ends in a 0 and starts in a 0

So a $\in$ $\{$w | w starts and ends on the same symbol$\}$

So $\{$w | w contains an equal number of occurrences of the substrings 01 and 10$\}$ $\subset$ $\{$w | w starts and ends on the same symbol$\}$

Other Direction

Prove $\{$w | w starts and ends on the same symbol$\}$ $\subset$ $\{$w | w contains an equal number of occurrences of the substrings 01 and 10$\}$

Let a $\in$ $\{$w | w starts and ends on the same symbol$\}$

So a starts and ends with the last symbol.

WLOG let a start and end in 0

Either there are no 1 in a, in which case there are 0 appearances of both 10 and 01 so a $\in$ A, or there is atleast a 1 in a

Since a starts and ends on the same symbol, for every appearance of the substring 01, when the current charactor becomes a 1 from a 0, there must eventually be an occurance of 10, otherwise a couldn't end in 0.

Since for every appearance of 01 there is a 10 associated with it, there are an equal number of 01 and 10 substrings in a

So a $\in$ A

So $\{$w | w starts and ends on the same symbol$\}$ $\subset$ $\{$w | w contains an equal number of occurrences of the substrings 01 and 10$\}$

Since both directions hold up, $\{$w | w starts and ends on the same symbol$\}$ = $\{$w | w contains an equal number of occurrences of the substrings 01 and 10$\}$

Proof

Prove that L(M) = A.

Let B = {w | w starts and ends on the same symbol}

First Direction

Prove that L(M) $\subset$ A

FSOC let L(M) $\not\subset$ A

So $\exists b \in L(M)$ such that $b \notin A$

Since $b \notin A$, $b \notin B$ by Claim 1

Since $b \notin B$, b must start and end with different symbols.

WLOG let b start with 0 and end with 1

Since b starts with 0, after the 0 is passed, the current state would be q_0.

Since the current state is q_0, everytime a 1 is encountered the state becomes q_01.

Since b ends with 1, the last state is q_01.

This contradicts the fact that q_01 is not an accept state but b $\in$ L(M) from the assumption.

Since there is a contradiction when assumed false, L(M) $\subset$ A.

Since there is a finite automata M for which A is the recognized language, A is a regular language.

Other direction

Prove A $\subset$ L(M)

FOSC let A $\not\subset$ L(M)

So $\exists b \in A$ such that $b \notin L(M)$

Since b $\notin$ L(M), after it is fed through M, the ending state is either q_01 or q_10.

WLOG let the ending state be q_01

Since the ending state is q_01, the first symbol in b must be 0 and the last one must be 1.

Since the first symbol in b is 0 and the last one is 1, the first and last symbols of b are not equal.

So $b \notin B$.

This contradicts Claim 1 since $b \in A$ but $b \notin B$ and by Claim 1, A = B

Since there is a contradiction when assumed false, $A \subset L(M)$

Since A $\subset$ L(M) and L(M) $\subset$ A, L(M) = A.

4 Problem 4

For any string $w = w_1 w_2 w_n$, the reverse of w, written as w^R, is the string w in reverse order, $w_n \quad w_2 w_1$. For any language A, let A^R = $\{w^R | w \in A\}$. Show that if A is regular, so is A^R

Proof:

Claim 1

If A is regular and it has only 1 accept state, then A^R is regular.

PROOF by construction: Since A is regular, it has a FA, $M = \{Q_1, \mathcal{E}, \delta, q_1, F_1\}$, that recognizes it, so $L(M) = A$

Since there is only one accept state, $|F_1| = 1$. Let f_1 be the only value in F_1

To construct a FA that recognizes A^R, we modify M into a NFA $M^R = \{Q_1, \mathcal{E}, \delta', f_1, \{q_1\}\}$ where δ' contains the flipped version of every transition. So if there was a transition from q_a to q_b in δ, there is a transition with the same requirement fom q_b to q_a in δ'.

Now we just need to show that $L(M^R) = A^R$.

Let the sequence of states M visits for input w be $p_1, p_2, ..., p_{n+1}$

$w^R \in A^R <==>$

$w \in A <==>$

$w \in L(M^R) <==>$

$p_{n+1} \in F_1 and p_1 = q_1 <==>$

$p_{n+1} = f_1 <==>$

$p_{n+1}...p_2, p_1$ is a valid sequence in $M^R <==>$

$w \in L(M^R)$

So $A^R = L(M^R)$.

Since an NFA recognizes the language, A^R is a regular language.

General Proof

I will prove that if A is regular, then A^R is regular.

I will prove so using induction on the number of accepted states in the finite automata M that recognizes A.

Base Case

n = 1, this was proven in claim 1

Induction Step

IH: $\exists k \in \mathcal{N}$ such that for all recognized languages A with a finite automata M that recognizes it and has k accepted states, A^R is a recognized language.

Lets look at when M that has $k+1$ accepted states.

let $F = \{a_1, a_2 ..., a_{n+1}\}$ where a_i is the ith accepted state.

So $M = \{Q, \mathcal{E}, \delta, q_0, F\}$

Lets divide this into two languages C and D where C is the language recognized by $M_1 = \{Q, \mathcal{E}, \delta, q_0, F\ \{a_{n+1}\}\}$ and D is the language recognized by $M_2 = \{Q, \mathcal{E}, \delta, q_0, \{a_{N+1}\}\}$.

So $C \cup D = A$ because D contains all strings that end on a_{n+1} when put through M, and C contains all strings that end in anything besides a_{n+1} that is accepted. The actual proof is shown below.

Let $a \in C \cup D$, so $a \in C$ or $a \in D$.

If $a \in C$ then a, when run through M, ends on an accept state, So $a \in A$

If $a \in D$ then a, when un through M, ends on a_{n+1} which is an accept state, so $a \in A$

Now let $b \in A$, so when run through M, the last state is a state in the set of accept states.

If the last state, when b was run through M is a_{n+1}, then $b \in D$ so $b \in C \cup D$

If the last state, when b was run through M is not a_{n+1}, then $b \in C$ since b had to land on an accept state, so $b \in C \cup D$

Since $C \cup D = A$, $C^R \cup D^R = A^R$. This is proven below:

Let $a^R \in C^R \cup D^R$ be an arbitrary element.

so $a^R \in C^R$ or $a \in D^R$

If $a^R \in C^R$, $a \in C$ so $a \in C \cup D$ so $a \in A$ since $C \cup D = A$ so $a^R \in A^R$

If $a^R \in D^R$, $a \in D$ so $a \in C \cup D$ so $a \in A$ since $C \cup D = A$ so $a^R \in A^R$

so either way, $a^R \in A^R$, so $C^R \cup D^R \subset A^R$

Now let $a^R \in A^R$ be an arbitrary element.

Since $a^R \in A^R$, $a \in A$

So $a \in C \cup D$ since $C \cup D = A$

So $a \in C$ or $a \in D$

If $a \in C$ then $a \in C^R$ so $a \in C^R \cup D^R$

If $a \in D$ then $a \in D^R$ so $a \in C^R \cup D^R$

so $A^R \subset C^R \cup D^R$

Since $A^R \subset C^R \cup D^R$ and $C^R \cup D^R \subset A^R$, $C^R \cup D^R = A^R$

Since C and D are defined of be the languages recognized by automata, they are regular languages.

Since C is a regular language and the number of accept states in M_1 is n, by IH C^R is a regular language.

Since D is a regular language and the number of accept states in M_2 is 1, by claim 1 D^R is a regular language.

Since C^R and D^R are regular languages, $C^R \cup D^R = A^R$ and regular languages are closed under union, A^R is a regular language.

So by induction for every regular language A whose finite automata has $n >= 1$ accept states, A^R is a regular language.