More Fun With Automata
Homework 2, CS500, Fall 2014

Aaron Gonzales,
group 16

January 29, 2015

Ex 7

> **Show that if L is regular then L^* is regular. Why does it not suffice to use the fact that the regular languages are closed under concatenation and union?**

Answer:

We know that $L* = \epsilon \cup L \cup LL \cup \ldots$, the enumerated set of all possible combinations of strings made up of the language L. We also know that for a language to be regular, there must be some DFA or NFA that can accept it.

If we let M_L be the machine that recognizes langague L, then let M_{L^*} be the machine that recognizes L^*. To construct M_{L^*}, add a new start state S' that is also an accepting state to M_L. Connect S' to M_L with an ϵ transition and add another ϵ edge out of M_L such that it connects to S'. This machine can accept any possible combinatins of valid instances of L, showing that L^* is indeed regular.

Ex 8

> **Given a string w , let w^R denote w written in reverse. Given a language L, let $L^R = \{w^R | w \in L\}$. Prove that L is regular if and only if L^R is regular. Why is this harder to prove with DFAs?**

Answer:

We know that a language is regular if there is a DFA that can accept it. This tells us that in order to prove W^R's regularity, there must be some DFA that can accept it. Generally put, we can map our current DFA to a DFA that can accept the new reversed language by modifying the start state, accepting states, and transition functions as follows, which will be formalized below.

Proof. Let $M = \{S, A, s^0, S^{yes}, \delta\}$ be the machine that recognizes L. Let $M' = S', A', S^{0'}, S^{yes'}, \delta*'$ be the NFA that recognizes L^R, constructed as below:

Let $S_{new} \in S$ be a new start state and connect it to all $s \in S^{yes}$ with an ϵ label. Make S^0 into an $S^{yes'}$. Make $s \in S^{yes} \to S'$ (make accepting states into normal states). Make $S^0 \to S^{yes'}$ (make the original intial state into the sole accepting state.

This new NFA will only accept langagues that begin in the first char of the w^R and end in the last char of w^r. As we have constructed an NFA that recognizes this L^R, we can deduce that that L^R is regular, meaning that $L \Leftrightarrow L^R$. □

The NFA epsilon ability allows us to more easily create new NFAs from exisisting automata.

Ex 12

> **Given a language L, let $L_{1/2}$ denote the set of words that can appear as first halves of words in L:**
>
> $$L_{1/2} = \{x | \exists y : |x| = |y| \text{ and } xy \in L\}$$
>
> **where $|w|$ denotes the length of a word w . Prove that if L is regular, then $L_{1/2}$ is regular. Generalize this to $L_{1/3}$, the set of words that can appear as middle thirds of words in L:**
>
> $$L_{1/3} = \{y | \exists x, z : |x| = |y| = |z| \text{ and } xyz \in L\}$$

Answer:

We know that $|x| = |y|$ and must show that the set of words in $L_{1/2}$ can be represented by a DFA to be proved regular. FA's limit us in that we cannot go back in time or record with external memory, and we may not know how large a word w is before it is read. As such, we must (A) build a FA that allows us to give $|x| = |y|$ and allows us to verify that (B) $xy \in L$. This is complicated but not impossible.

Chris Moore's hint about River Song and the Doctor is a reference to Dr. Who, in which the characters are moving toward each other in time, one going forward and the other going backward. Given that hint, let us construct a FA that allows us to accomplish A and B, mostly given by the product construction of two FAs.

Proof. Let us define two FAs:

$$M = \{Q, \Sigma, \delta, S_0, S^{yes}\} \tag{1}$$

$$M^R = \{Q^R, \Sigma, \delta^R, S^{yes^R}\} \tag{2}$$

$$Q^R = Q \cup \{S_0^R\} \tag{3}$$

M is the DFA which recognizses L and M^R is the DFA that recognizses L^R. We have seen that the reverse operator is closed under regularity. Q^R is the union of states of M and acepting states of L^R.

□

Ex 9

A for-all NFA is one such that $L(M)$ is the set of strings where every computation path ends in an accepting state. Show how to simulate an for-all NFA with a DFA, and thus prove that a language is recognized by some for-all NFA if and only if it is regular.

Answer:

Ex 13

Show that if $u_L v$, then $ua_L va$ for any $a \in A$.

Answer:

blah blah $\wr$ blah blah blah

$$W|\star \to \wr \sum_{i}^{x}$$

Ex 14

Describe the equivalence classes of the three languages from Exercise 2. Use them to give the minimal DFA for each language, or prove that the DFA you designed before is minimal.

Answer:

Ex 20

Consider the language

$$L_{a=b,c=d} = \{w \in \{a, b, c, d\} \star \mid \#_a(w) = \#_b(w) \text{ and } \#_c(w) = \#_d(w)$$

What are its equivalence classes? What does its minimal infinite-state machine look like?

Answer: