Automata Tutorial

May 28, 2018

Definition 1 A Deterministic Finite Automaton(DFA) is a structure $M = (Q, \Sigma, \delta, s, \mathbb{F})$ where

$$
\begin{aligned}
Q - &\text{ is a finite non empty set of states} \\
\Sigma - &\text{ is a finite non empty set of input alphabets} \\
\delta : Q \times \Sigma \to Q - &\text{ transition function} \\
s \in Q - &\text{ is the start state} \\
\mathbb{F} \subseteq Q - &\text{ is the set of accept states}
\end{aligned}
$$

The transition function δ can be extended to another function $\hat{\delta} : Q \times \Sigma^* \to Q$ which would capture the idea of transitions across a word. And, $\mathbb{L}(M) = \{ \ w \in \Sigma^* \mid \hat{\delta}(s, w) \in \mathbb{F} \ \}$ is the language of the automaton $\mathbb{M}$.

Question Set 1 For our purposes $\Sigma = \{0, 1\}$

- Write out an inductive definition of $\hat{\delta}$ where you'd induct over the length of the word $w \in \Sigma^*$.
- Give the DFA definition for the machine whose language is $\{x \in \Sigma^* \mid x$ is the binary representation of 12 $\}$. How many states did you need? Can you do it in fewer number of states?

Construct DFA for the following languages over Σ as specified above.

- $\{x \in \Sigma^* \mid x$ is a binary representation of an even number $\}$
- $\{x \in \Sigma^* \mid$ length of x is even $\}$
- $\{x \in \Sigma^* \mid$ the decimal representation of x is divisible by 3 $\}$

- $\{x \in \Sigma^* \mid$ the length of x is divisble by $3\}$
- $\{x \in \Sigma^* \mid x$ contains 011 as a substring $\}$
- $\{x \in \Sigma^* \mid x$ does not contain 011 as a substring $\}$
- $\{x \in \Sigma^* \mid$ the decimal representation of x is a power of 2 $\}$
- $\{x \in \Sigma^* \mid$ the decimal representation of x is divisible by 6 $\}$
- $\{x \in \Sigma^* \mid$ the length of x is divisible by 2 or 3 $\}$

Definition 2 A Non-Deterministic Finite Automaton(NFA) is a structure A Deterministic Finite Automaton(DFA) is a structure $M = (Q, \Sigma, \delta, S, F)$ where the only definitional changes from the DFA are :

$$\delta : Q \times \Sigma \rightarrow 2^Q - \text{is the transition function}$$
$$S \subseteq Q - \text{is the set of start states}$$

Question Set 2

- Come up with an inductive definition of $\hat{\delta}$ from the notion of δ to capture connectivity of two states via words.
- Come up with the definition of the language accepted by a NFA, using the notion of $\hat{\delta}$ as defined above.

Construct NFA for the following languages where $\Sigma = \{a, b\}$

- $\{x \in \Sigma^* | \ x$ ends with three consecutive a $\}$
- $\{x \in \Sigma^* | \ x$ has odd number of a$\}$
- $\{x \in \Sigma^* | \ x$ has the substring aab $\}$
- $\{x \in \Sigma^* | \ x$ has length divisible by 3 $\}$

Question 3 If A,B be two subsets of Σ^* then A$\odot$B=$\{ab \mid a \in A, b \in B\}$. Let M_1, M_2 be two DFAs, then come up with the DFA to accept the language $\mathbb{L}(M_1) \cup \mathbb{L}(M_2)$ and $\mathbb{L}(M_1) \odot \mathbb{L}(M_2)$. Why is the second one difficult to do? What ideas do you have to circumvent the bottleneck?

Question 4 If M is a DFA, suggest algorithms to check if a) $\mathbb{L}$(M) $= \phi$ and b) if $\mathbb{L}$(M) is finite. Analyse the running times of the algorithms.

Question 5 Suppose there are DFA $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1), M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$, with $|Q_2| < |Q_1|$ and $\mathbb{L}(M_1) = \mathbb{L}(M_2)$, then we say that M_2 is an equivalent DFA of lesser size.
Consider DFA with states $\mathbb{Z}_5$ and $\Sigma = \{0, 1\}$ with start state and final state both being 0, and transition function being,

$$\delta(q, i) = (q^2 + i) \ (mod \ 5)$$

for $q \in \mathbb{Z}_5, i \in \Sigma$. Can you give an Equivalent DFA of lesser size? What about trying to find the minimal such DFA which will accept the same language as the automaton?

Question 6 Let $A, B \subseteq \Sigma^*$ be two languages that have NFAs to accept them. Let's define

$$A/B = \{x \in \Sigma^* \mid \exists y \in B, xy \in A\}$$

. Show that there is a NFA that will accept A/B.

Question 7 Construct an NFA over a unary alphabet that rejects some string, but the length of the shortest rejected string is strictly more than the number of states. What is the language accepted by your NFA?

Question 8 Give a family of languages E_n, where each E_n can be recognised by an n-state NFA, but requires at least c^n states on a DFA for some constant $c > 1$.

Definition 3 A Turing Machine is a 7-tuple, $T = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ where the it resembles the definitions for finite automata and the new definitional additions are

$$\begin{aligned} \Gamma \subseteq \Sigma - & \text{ is the tape alphabet} \\ \delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\} - & \text{ is the transition function} \\ q_{\text{accept}} \in Q - & \text{ is the accept state} \\ q_{\text{reject}} \in Q - & \text{ is the reject state} \end{aligned}$$

- How do we define the languages accepted by this machine so constructed?
- This definition is for a single tape turing machine, how do we go from here to talk about a multi tape turing machine?
- Prove that you do not add extra power by introducing more tapes to the turing machine. Basically, every multitape turing machine has an equivalent single tape turing machine that can decide the same language as the multitape one.

Question Set 10 How many distinct two letter DFAs are there that have 2 states? What are the languages accepted by each of them? Same question but for NFAs.