Lecture 3: Two-way Finite State Automaton

Instructor: Ketan Mulmuley
Scriber: Yuan Li

January 13, 2015

1 Two-way Finite State Automaton

Recall that a two-way deterministic finite state automaton (2-DFA) $M = (Q, \Sigma, \delta, q_0, F)$, where $q : Q \times \Sigma \to Q \times \{L, R\}$ is the transition function. For example, $\delta(q, a) = (p, L)$ means, at current state p if the symbol is a, then the next state is p and the head will move left. M accepts a string w if it stops at a final state, and the head is on the right-most position.

Theorem 1.1. *2-DFA is equivalent to NFA (and thus DFA), that is, if a language is accepted by some 2-DFA, then there exists an NFA accepting the same language.*

Proof. Given a 2-DFA $M = (Q, \Sigma, \delta, q_0, F)$, we will construct an NFA $M' = (Q', \Sigma, \delta', q_0', F')$ such that $L(M) = L(M')$.

Before getting into the construction, we need the concept *valid crossing sequence*. As the following picture illustrates, the head first moves right in state q_1, working on the right side of the tape, and then moves left in state q_2, working on the left side of the tape, and then moves right in state q_3, ..., finally the head moves left in state q_k.

Observe that the followings are true.

- The length k is odd.
- The same state does not repeat at any two odd locations or any two even locations. (Otherwise, the machine will run into a loop.) This implies the length $k < 2|Q|$.

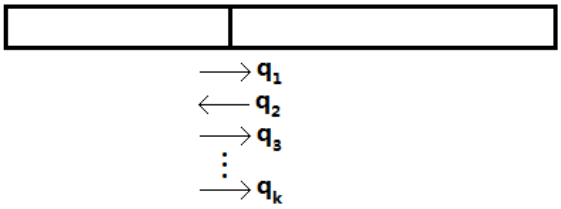

Figure 1: valid crossing sequence

Let Q' be the set of all valid crossing sequences, where $|Q'| \leq |Q|^{2|Q|}$, $q_0 = [q_0]$ and $F' = \{[p] : p \in F)$. For the transition function $\delta' : Q' \times \Sigma \to Q'$,

$$\delta'([q_1, \ldots, q_k], a) = [p_1, \ldots, p_l]$$

if and only if $[q_1, \ldots, q_k]$ is a-compatible with $[p_1, \ldots, p_l]$, that is, $\delta(q_1, a) = (p_1, R)$, $\delta(p_1, a) = (q_2, L), \ldots$, and $l = k$.

It is an NFA instead of DFA because for any given crossing sequence $[q_1, \ldots, q_k]$, there can be multiple $[p_1, \ldots, p_l]$ that is a-compatible with $[a_1, \ldots, a_k]$. □

Note that in the above construction, $|Q'| = |Q|^{2|Q|}$. Converting to DFA, the number of states would be

$$2^{|Q'|} = 2^{|Q|^{2|Q|}}.$$

If $|Q| = 10$, then the DFA will have $2^{10^{20}}$ states, which is greater than the number of particles in the universe. In that sense, the above theorem is a purely mathematical statement. Understanding the difference between polynomial and exponential is what complexity theory all about.

The following exercises are challenging, which says the exponential blow-up is necessary.

Exercise 1.2 (*)**.** *Construct a 2-DFA with $|Q|$ states such that there does not exist any equivalent NFA with the number of states $\leq |Q|^a$ for any constant $a > 0$.*

Exercise 1.3 (*)**.** *Construct a 2-DFA with $|Q|$ states such that there does not exist any equivalent DFA with $2^{|Q|^a}$ states for any constant $a > 0$.*

2 Lexical Analyzer

In ALGOL, an identifier has the following form (regular expression)

$$(letter)(letter + digit)^*$$

In Fortan,

$$(letter)(\epsilon + letter + digit)^*$$

Exercise 2.1. *Construct a DFA accepting the above regular expressions.*

In many compiler, the lexical analyzers are implemented using DFA.

3 Pumping Lemma

Pumping lemma says something about regular language, which is not as great as its name.

Lemma 3.1 (Pumping Lemma)**.** *If $L \subseteq \Sigma^*$ is a regular language, then there exists a constant n such that if $z \in L$ and $|z| > n$, then there exists $u, v, w \in \Sigma^*$ such that $z = uvw, |uv| \leq n$ and $|v| \geq 1$ and $uv^i w \in L$ for any $i \geq 0$.*

Proof. Take a DFA $M = (Q, \Sigma, \delta, q_0, F)$ accepting L, and let $n = |Q|$. Let $z \in L$ be a string with length $|z| > n$. Consider the sequence of states in M accepting z. Since $|z| > |Q|$, there must exist a state which has been visited more than once; let p be the *first* such state. Let u be the string from q_0 to p, and let v be the *first* loop from p to itself, and let w be the rest.

It is easily checked that all the properties in the lemma are satisfied. □

Let us see a few applications of pumping lemma.

Proposition 3.2. *$L = \{0^l 1^l : l \geq 0\}$ is not regular.*

Proof. Suppose to the contrary that L is regular. Let n be as in the pumping lemma. Let $l > n$.

Let $z = 0^l 1^l \in L$. By the pumping lemma, $0^l 1^l = uvw$, where $|uv| \leq n$, which implies both u and v are consecutive 0's. By the lemma, $uv^2 w \in L$, which is a contradiction, because $uv^2 w$ has more 0's than 1's. □

Proposition 3.3. *$L = \{0^{i^2} : i \geq 0\}$ is not regular.*

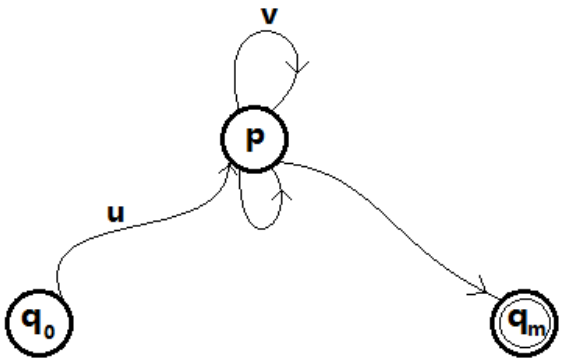

Figure 2: path accepting z

Proof. Suppose to the contrary that L is not regular. Let n be as in the pumping lemma. Consider

$$z = \underbrace{00\ldots0}_{(n+1)^2 \text{ times}} .$$

By the pumping lemma, $z = uvw$ where $|uv| \leq n$ and $|v| \geq 1$, and $uv^2w \in L$. However,

$$|uv^2w| = (n+1)^2 + |v| \leq (n+1)^2 + n < (n+2)^2,$$

which is a contradiction! □

Exercise 3.4. *Prove $L = \{0^p : p \text{ is a prime}\}$ is not regular.*

4 Properties of Regular Language

Proposition 4.1. *If L is regular, then $\overline{L} = \Sigma^* \setminus L$ is also regular.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F \subset Q)$ be a DFA such that $L = L(M)$, and without loss of generality assume M will not get stuck. Let $M' = (Q, \Sigma, \delta, q_0, Q \setminus F)$, that is, flip accept states to reject states, and reject states to accept states. □

Proposition 4.2. *If L_1, L_2 are regular, then $L_1 \cup L_2$ is also regular.*

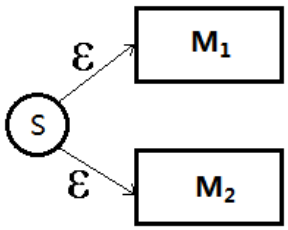

Figure 3: ϵ-NFA accepting $L_1 \cup L_2$

Proof. Let M_1, M_2 be the NFAs accepting L_1, L_2 respectively. Construct the following NFA with ϵ-moves, which accepts $L_1 \cup L_2$. □

Proposition 4.3. *Let L_1, L_2 are regular, then $L_1 \cap L_2$ is regular.*

Proof. Since L_1, L_2 are regular, we have $\overline{L_1}, \overline{L_2}$ are regular. Then

$$\overline{L_1} \cup \overline{L_2} = \overline{L_1 \cap L_2}$$

is regular, which implies $L_1 \cap L_2$ is regular. □