Lecture #21: Chomsky Hierarchy I — Regular Languages
Exercises and Review

Questions for Review

1. What is a ***right-regular grammar***? What is a ***regular language*** (as defined in the preparatory reading)?
2. What is a ***deterministic finite automaton***, and how is the ***language of a deterministic finite automaton*** defined? How are the regular languages, and the languages of deterministic finite automata, related?
3. What is a ***left-regular grammar***? How are the regular languages, and the languages of left-regular languages, related?
4. Describe a result (which you ideally knew about, before this course) which can be used to prove that a language is ***not*** regular.
5. Describe one more computational problems concerning regular languages and approaches to solving them.

Problems To Be Solved

1. Consider an alphabet Σ. Recall that if

   $$\omega = \alpha_1\alpha_2\ldots\alpha_n$$

   is a string with length n in $\Sigma^\star$ (so that $\alpha_1, \alpha_2, \ldots, \alpha_n \in \Sigma$), then the ***reversal*** of ω is the string

   $$\omega^R = \alpha_n\ldots\alpha_2\alpha_1$$

   obtained by listing the symbols in ω in reverse order.

   If $L \subseteq \Sigma^\star$ then one can define the ***reversal*** of L to be the language

   $$L^R = \{\omega^R \mid \omega \in L\} \subseteq \Sigma^\star.$$

Prove, for every language $L \subseteq \Sigma^\star$, that if L is a regular language then L^R is a regular language as well — *without* using Claim #2 in the required reading.

Note that, since $(L^R)^R = L$, for any language $L \subseteq \Sigma^\star$, it follows that L is a regular language ***if and only if*** L^R is a regular language, for every language $L \subseteq \Sigma^\star$.

Note: There are several ways to prove this. One way is recall that if L is a language then it is the language of some ***regular expression*** φ — and then describe (and sketch a proof of the correctness of) a process that can be used to convert φ into a regular expression $\widehat{\varphi}$, such that the language of $\widehat{\varphi}$ is the reversal of the language of φ.

2. Now consider a right-regular grammar $G = (V, \Sigma, \Pi, S)$. Let $G^R = (V, \Sigma, \Pi^R, S)$ be a phrase-structure grammar such that the productions in Π^R are obtained from the productions in Π as follows.

   - For all variables $A \in V$ and symbols $\sigma \in \Sigma$, if $A \to \sigma$ is a production in Π, then $A \to \sigma$ is also a production in Π^R.
   - For all variables $A, B \in V$ and all symbols $\sigma \in \Sigma$, if $A \to \sigma B$ is a production in Π, then $A \to B\sigma$ is a production in Π^R.
   - For all variables in A, if $A \to \lambda$ is a production in Π then $A \to \lambda$ is also a production in Π^R.
   - There are no other productions in Π^R.

   Note that, while G is a ***right-regular grammar***, G^R is a ***left-regular*** grammar.

   (a) Let $A, B \in V$ and let $\omega \in \Sigma^\star$. Prove that $A \Rightarrow_G^\star \omega B$ (that is, ωB can be derived from A in G) if and only if $A \Rightarrow_{G^R}^\star B\omega^R$ (that is, $B\omega^R$ can be derived from A in G^R).

   (b) Use this to show that if $L = L(G)$ then $L^R = L(G^R)$.

3. Use the above results to prove Claim #2 in the required reading: For all languages $L \subseteq \Sigma^\star$ (over some alphabet Σ), L is a regular language if and only if L is the language of a left-regular grammar.

4. Let $M = (Q, \Sigma, \delta, q_0, F)$ A state $r \in Q$ is ***reachable*** from a state $q \in Q$ if there exists a string $\omega \in \Sigma^\star$ such that

$$\delta^\star(q, \omega) = r.$$

   A state $r \in Q$ is ***k-reachable*** from a state $q \in Q$ if there exists a string $\omega \in \Sigma^\star$ such that $\delta^\star(q, \omega) = r$, and $|\omega| \leq k$.

   Now, let $q \in Q$.

   - For every non-negative integer k, let $S_q^{(k)}$ the the set of states in Q that are k-reachable from q.
   - Let S_q be the set of states in Q that are reachable from q.

Note that $S_q^{(k)} \subseteq S_q$ for all states $q \in Q$ and for all non-negative integers k

(a) Prove that $S_q^{(0)} = \{q\}$ for every state $q \in Q$.

(b) Prove that

$$S_q^{(k+1)} = S_q^{(k)} \cup \{\delta(r, \sigma) \mid r \in S_q^{(k)} \text{ and } \sigma \in \Sigma\}$$

for every state $q \in Q$ and for every non-negative integer k.

(c) Prove, for a state $q \in Q$ and non-negative integer k, that if $S_q^{(k)} = S_q^{(k+1)}$ then $S_q^{(k)} = S_q$.

(d) Use the above to design an algorithm that can be used to compute S_q, for a state $q \in Q$, and sketch a proof that your algorithm is correct.

5. Let $M_1 = (Q_1, \Sigma, \delta_1, q_{0,1}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2 q_{0.2}.F_2)$ be deterministic finite automata with the same alphabet, Σ. One can apply a ***product construction*** to produce a variety of deterministic finite automata that share the following properties.

- The set of states is $Q_1 \times Q_2 = \{(q_1, q_2) \mid q_1 \in Q_1 \text{ and } q_2 \in Q_2\}$.
- The alphabet is Σ, just as it is for M_1 and M_2.
- The transition function is a function $\delta_{1,2} : (Q_1 \times Q_2) \times \Sigma \to (Q_1 \times Q_2)$ such that, for all $q_1, r_1 \in Q_1$, $q_2, r_2 \in Q_2$ and $\sigma \in \Sigma$, if $\delta_1(q_1, \sigma) = r_1$ and $\delta_2(q_2, \sigma) = r_2$, then

$$\delta_{1,2}((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) = (r_1, r_2).$$

(a) Prove that $\delta^\star_{1,2}((q_1, q_2), \omega) = (\delta^\star_1(q_1, \omega), \delta^\star_2(q_2, \omega))$ for all states $q_1 \in Q_1$ and $q_2 \in Q_2$, and for all strings $\omega \in \Sigma^\star$ as well.

(b) Describe how to define the set $F_{1,2}$ os a deterministic finite automaton with this set of states, start state, and transition, so the language of the DFA is the following.

(i) $L(M_1) \cup L(M_2)$

(ii) $L(M_1) \cap L(M_2)$

(iii) $L(M_1) \setminus L(M_2)$