Problem Set 5 Solutions

Problem 1. Specify a CFG for the language

\[ L = \{ x \in \{\text{bass, chicken, carp, turkey}\}^* : x \text{ contains as much fish as fowl} \} \]

(meaning that the number of occurrences in \( x \) of substrings bass and carp should be at least the number occurrences in \( x \) of substrings chicken and turkey. Make your CFG as simple to understand as you can.

A CFG for this language is given by

\[ S \rightarrow F \rightarrow \text{Fish} \rightarrow \text{bass} | \text{carp} \]
\[ S \rightarrow F \rightarrow \text{Fish} \rightarrow \text{bass} | \text{carp} \]
\[ X \rightarrow \text{chicken} \rightarrow \text{turkey} \]

Problem 2. Prove that every regular language is context free. Do this by showing how to convert a DFA \( M = (Q, \Sigma, \delta, q_0, F) \) into a CFG \( G = (V, \Sigma, R, S) \) of roughly the same size.

Given the DFA \( M = (Q, \Sigma, \delta, q_0, F) \) we construct the CFG \( G = (V, \Sigma, R, S) \) by asserting that

- \( V = Q \)
- For \( p, q \in Q \) and \( a \in \Sigma \), put
  \[ p \rightarrow aq \in R \text{ id } \delta(p, a) = q; \text{ and put } p \rightarrow \epsilon \in R \text{ if } p \in F. \]
- \( S = q_0 \)

To show \( L(M) = L(G) \):

- First we show that if \( x \in L(M) \) then \( x \in L(G) \), because we can derive \( x \) in \( G \) as follows:
  - If \( x = \epsilon \in L(M) \) then derive \( x \) by \( q_0 \Rightarrow \epsilon \).
  - If \( x = a_1 \cdots a_n \in L(G) \), with \( a_i \in \Sigma \), then we derive \( x \) by \( q_0 \Rightarrow a_1 \delta^*(q_0, a_1) \Rightarrow a_1a_2\delta^*(q_0, a_1a_2) \Rightarrow a_1a_2a_3\delta^*(q_0, a_1a_2a_3) \Rightarrow \cdots \Rightarrow a_1a_2a_3 \cdots a_n \Rightarrow a_1a_2a_3 \cdots a_n \epsilon = x. \)
- Next we show that if \( x \in L(G) \) then \( x \in L(M) \). For \( x \in L(G) \) means there’s a derivation of \( x \) from \( q_0 \) and, because of the limited rules in our CFG, the derivation can only look like \( q_0 \Rightarrow a_1q_1 \Rightarrow a_1a_2q_2 \Rightarrow a_1a_2a_3q_3 \Rightarrow \cdots \Rightarrow a_1a_2a_3 \cdots a_nq_n \Rightarrow a_1a_2a_3 \cdots a_n \epsilon = x \) where each \( a_i \in \Sigma \) and \( q_i \in Q \). But then \( x \in L(M) \), for \( q_0q_1 \cdots q_n \) is a path in the DFA from the start state to the final state labeled by \( x \).

Problem 3. Prove that every regular language is context free. Do this by showing how to convert a regular expression \( \alpha \) into a CFG \( G = (V, \Sigma, R, S) \) of roughly the same size.

By induction. The regular expressions \( a \ (a \in \Sigma), \varepsilon, \) and \( \emptyset \) have CFGs \( S \rightarrow a, S \rightarrow \varepsilon, \) and \( S \rightarrow \varepsilon \), respectively. Now suppose we have built CFGs \( G_\alpha = (V_\alpha, \Sigma, S_\alpha, R_\alpha) \) and \( G_\beta = (V_\beta, \Sigma, S_\beta, R_\beta) \) for regular expressions \( \alpha \) and \( \beta \). Rename symbols, if necessary, so that \( V_\alpha \) and \( V_\beta \) are disjoint. Then we can build a CFG for \( (\alpha \circ \beta) \) by \( (V_\alpha \cup V_\beta \cup \{S\}, \Sigma, S, R_\alpha \cup R_\beta \cup \{(S, S_\alpha S_\beta)\}) \). Similarly, we can build a CFG for \( (\alpha \cup \beta) \) by \( (V_\alpha \cup V_\beta \cup \{S\}, \Sigma, S, R_\alpha \cup R_\beta \cup \{(S, S_\alpha), (S, S_\beta)\}) \). And we can build a CFG for \( (\alpha)^* \) by \( (V_\alpha \cup \{S\}, \Sigma, S, R_\alpha \cup \{(S, S_\alpha S_\alpha), (S, \varepsilon)\}) \).