Context-free languages are closed under intersection with regular languages

Stephen Checkoway

February 24, 2014

The proof of the main theorem below is greatly simplified by the introduction of new notation.

Definition. For a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is a DFA, define the function $\delta^* : Q \times \Sigma^* \to Q$ by

$$\begin{aligned} \delta^*(q, \varepsilon) &= q \\ \delta^*(q, aw) &= \delta^*\big(\delta(q, a), w\big) \qquad \text{for } a \in \Sigma,\ w \in \Sigma^*. \end{aligned}$$

In essence, starting from state q, when M reads the string w, it ends up in state $\delta^*(q, w)$. Note that $w \in L(M)$ if and only if $\delta^*(q_0, w) \in F$.

Theorem. *The intersection of a context-free language L_1 and a regular language L_2 is context-free.*

For any CFG $G = (V, \Sigma, R, S)$ in Chomsky normal form (CNF) that does not generate ε and a DFA $M = (Q, \Sigma, \delta, q_0, \{q_f\})$ that has exactly one accept state, we can construct a new CFG $G' = (V', \Sigma, R', S')$, also in CNF where

$$\begin{aligned} V' &= \{\langle q, A, r\rangle \mid A \in V \text{ and } q, r \in Q\}, && (1) \\ S' &= \langle q_0, S, q_f\rangle, && (2) \\ R' &= \{\langle q, A, r\rangle \to t \mid A \to t \in R,\ t \in \Sigma \cup \{\varepsilon\}, \text{ and } \delta(q, t) = r\} \cup && (3) \\ &\quad\ \{\langle q, A, r\rangle \to \langle q, B, s\rangle\langle s, C, r\rangle \mid A \to BC \in R \text{ and } q, r, s \in Q\}. && (4) \end{aligned}$$

The new grammar G' is clearly in CNF since each rule is either $\langle \textit{variable}\rangle \to \langle \textit{terminal}\rangle$ from (3) or $\langle \textit{variable}\rangle \to \langle \textit{variable}\rangle\langle \textit{variable}\rangle$ from (4).

The intuition behind these variables is that $\langle q, A, r\rangle$ generates the strings w that are generated by A in G such that when M reads w starting from state q, it ends in state r. We make that more precise and prove that it is true with the following lemma.

Lemma. *For each $\langle q, A, r\rangle \in V'$, $\langle q, A, r\rangle \overset{*}{\Rightarrow} w$ iff $A \overset{*}{\Rightarrow} w$ and $\delta^*(q, w) = r$.*

Proof. We can prove this by induction on the length of strings w. There are two cases to consider.

1. Base case: $w = a$ for some $a \in \Sigma$. Since G' is in CNF, the derivation of a terminal happens in a single step. Thus, $\langle q, A, r\rangle \overset{*}{\Rightarrow} a$ iff $\langle q, A, r\rangle \Rightarrow a$ iff $A \Rightarrow a$ and $\delta(q, a) = r$ iff $A \overset{*}{\Rightarrow} a$ and $\delta^*(q, a) = r$. The last step is an "iff" for the same reason the first is: G is in CNF.

2. Inductive case: $|w| = n > 1$. Deriving a string of length $n > 0$ from a grammar in CNF takes $2n - 1$ steps. Since $n > 1$, this first step *must* yield two variables. Therefore, $\langle q, A, r\rangle \overset{*}{\Rightarrow} w$ iff

   $$\langle q, A, r\rangle \Rightarrow \langle q, B, s\rangle\langle s, C, r\rangle \overset{*}{\Rightarrow} w \qquad \text{for some } s \in Q \tag{5}$$

   iff $A \Rightarrow BC$.

   Now we can apply the inductive hypothesis twice since each variable in the middle of (5) must derive a string of length strictly smaller than n. In particular, neither variable may derive ε because only the start variable in a CNF grammar may derive the empty string and the start variable may not appear in the right hand side of any rule. Thus, by the inductive hypothesis, $\langle q, B, s\rangle \overset{*}{\Rightarrow} w_1$ and $\langle s, C, r\rangle \overset{*}{\Rightarrow} w_2$, iff $B \overset{*}{\Rightarrow} w_1$, $\delta^*(q, w_1) = s$, $C \overset{*}{\Rightarrow} w_2$, and $\delta^*(s, w_2) = r$. Since $w = w_1w_2$,

   $$\begin{aligned}\delta^*(q, w) &= \delta^*\big(\delta^*(q, w_1), w_2\big)\\ &= \delta^*(s, w_2)\\ &= r.\end{aligned}$$

   Since $A \Rightarrow BC$, $A \overset{*}{\Rightarrow} w$.

   Putting this all together, we have $\langle q, A, r\rangle \overset{*}{\Rightarrow} w$ iff $A \overset{*}{\Rightarrow} w$ and $\delta^*(q, w) = r$. □

In particular, the strings generated by $\langle q_0, S, q_f\rangle$ are precisely those strings generated by S which are accepted by M. All that remains to prove the theorem is to handle the cases where the DFA recognizing L_2 has zero accept states (i.e., $L_2 = \emptyset$), the DFA has more than 1 accept states, and where $\varepsilon \in L_1$.

Proof. If $L_2 = \emptyset$, then $L_1 \cap L_2 = \emptyset$ which is context-free.

Assume $L_2 \neq \emptyset$. It is an easy fact to prove that any nonempty, regular language is the union of finitely many regular languages each of which is recognized by a DFA with a single state.[1] Since context-free languages are closed under union, it suffices to prove the theorem for the case where L_2 is recognized by a DFA with a single accept state.

Let $G = (V, \Sigma, R, S)$ be a CFG in CNF which generates $L_1 \setminus \{\varepsilon\}$ and let $M = (Q, \Sigma, \delta, q_0, \{q_f\})$ be the DFA which recognizes L_2. Construct the new CFG G' according to the above construction. Now, $w \in L(G')$ iff $\langle q_0, S, q_f\rangle \overset{*}{\Rightarrow} w$. By

[1] To see this, consider a DFA which recognizes the original language. This DFA has $|F|$ accept states. Construct $|F|$ copies of the DFA, each of which has a single accept state. The union of the language recognized by each of these machines is the original language.

the lemma, this happens iff $S \overset{*}{\Rightarrow} w$ and $\delta^*(q_0, w) = q_f$. Hence $w \in L(G')$ iff $w \in L_1 \setminus \{\varepsilon\}$ and $w \in L_2$.

Finally, if $\varepsilon \in L_1 \cap L_2$, then we can add the rule $\langle q_0, S, q_f \rangle \to \varepsilon$ to G'. If we do this, G' is still in CNF. In particular, $\langle q_0, S, q_f \rangle$ never appears on the right hand side of a rule so all the introduction of this rule does is add ε to the language generated by G'. In either case, $L(G') = L_1 \cap L_2$. $\square$