

Non-CFLs

A. Last time

Last time, we proved the pumping lemma for context-free languages. Similar to how we use the pumping lemma for regular languages, we can use the pumping lemma for context-free languages to prove that a language is not context-free.

We have the “demon game” version of the contrapositive of the pumping lemma. To prove that a language L is not regular, we must win the following game.

1. The demon picks $k \geq 0$ (the pumping length).
2. You pick $z \in L$ such that $|z| \geq k$.
3. The demon picks $u, v, w, x, y \in \Sigma^*$ such that $uvwxy = z$, $|vx| > 0$, and $|vwx| \leq k$.
4. You pick $i \geq 0$.

You win the game if $uv^iwx^iy \notin L$.

We saw last time that the choice of i gets to depend on the choice of u, v, w, x, y .

Specifically, we proved that the language

$$A = \{a^n b^n c^n \mid n \geq 0\} \tag{1}$$

is not context-free. This required considering multiple cases corresponding to the different possible choices of u, v, w, x, y . In some cases, we picked $i = 0$ and in one case, we picked $i = 2$.

B. Intersecting a CFL and a regular language

Consider the language

$$B = \{x \in \{a, b, c\}^* \mid \#_a(x) = \#_b(x) = \#_c(x)\}. \tag{2}$$

This language *feels* like it should not be context-free. We can use the pumping lemma for CFLs to prove that it isn't, however there's an even easier method to proving that it's not CF that requires first proving a general theorem about the intersection of a CFL and a regular language.

Theorem 1. *If L_1 is a context-free language and L_2 is a regular language, then $L_1 \cap L_2$ is context-free. (In other words, the class of context-free languages is closed under intersection with a regular language.)*

The idea behind the proof is we're going to transform a CFG in Chomsky Normal Form (CNF) for language L_1 into a CFG for $L_1 \cap L_2$. The way we're going to do this is for each nonterminal A in the CFG for L_1 will become a bunch of nonterminals $A_{q,r}$ where q and r are states in the DFA for L_2 . The intention behind the construction is that if $A_{q,r} \xRightarrow{*} x \in \Sigma^*$, then $A \xRightarrow{*} x$ and $\hat{\delta}(q, x) = r$.

Proof. Let $G = (N, \Sigma, P, S)$ be a CFG in CNF whose language is L_1 and let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA whose language is L_2 . Construct a new CFG, $G' = (N', \Sigma, P', S')$ as follows.

Define $N' = \{A_{q,r} \mid A \in N \text{ and } q, r \in Q\} \cup \{S'\}$ where S' is a new start nonterminal.

Since G is in CNF, each rule must have one of three possible forms

$$\begin{array}{ll} S \rightarrow \varepsilon & S \text{ is the start nonterminal} \\ A \rightarrow BC & \text{for } A \in N, \text{ and } B, C \in N \setminus \{S\} \\ A \rightarrow t & \text{for } t \in \Sigma. \end{array}$$

We'll deal with each type of rule.

For each rule $A \rightarrow BC \in P$, add the $|Q|^3$ rules

$$A_{q,r} \rightarrow B_{q,s}C_{s,r} \quad \text{for } q, r, s \in Q$$

to P' . [Since our intention is for strings derived by $A_{q,r}$ to be the strings that cause the DFA to transition from state q to state r , this is saying that if $B \xRightarrow{*} y$ and $C \xRightarrow{*} z$, then $A_{q,r} \Rightarrow B_{q,s}C_{s,r} \xRightarrow{*} yz$ and $\hat{\delta}(q, y) = s$ and $\hat{\delta}(s, z) = r$ for some intermediate state s .]

For each rule $A \rightarrow t \in P$ and each pair of states $q, r \in Q$ such that $\delta(q, t) = r$, add the rule

$$A_{q,r} \rightarrow t$$

to P' .

For each $f \in F$, add the rule

$$S' \rightarrow S_{q_0,f}$$

to P' . These rules account for the fact that there can be multiple possible accepting states in the DFA M . [Strings derived from $S_{q_0,f}$ are precisely the strings x derivable from S —which is to say strings from L_1 —such that $\hat{\delta}(q_0, x) = f$. Although we haven't yet proved that.]

Finally, if $S \rightarrow \varepsilon \in P$ and $q_0 \in F$ (which is to say ε is in L_1 and in L_2), add the rule

$$S' \rightarrow \varepsilon$$

to P' .

The construction is complete. All that remains is to prove that it's correct. The proof depends on a lemma which I'll state and leave the proof as a simple exercise.

Lemma. *For all $A_{q,r} \in N'$ and $x \in \Sigma^* \setminus \{\varepsilon\}$, $A_{q,r} \xRightarrow{*}_{G'} x$ if and only if $A \xRightarrow{*}_G x$ and $\hat{\delta}(q, x) = r$.*

The lemma can be proved by strong induction on the length of the derived string x where the base case is strings of length 1 for which come from the rules $A_{q,r} \rightarrow t$ and the inductive case is longer strings which must come from the rules $A_{q,r} \rightarrow B_{q,s}C_{s,r}$.

We can complete the proof that $L(G') = L_1 \cap L_2$. First, note that if $\varepsilon \in L_1 \cap L_2$, then by construction $S' \rightarrow \varepsilon$ is a rule in G' so $\varepsilon \in L(G')$.

Second, consider nonempty strings x . We have the chain of if and only ifs,

$$\begin{aligned} x \in L(G') &\iff S' \Rightarrow S_{q_0,f} \xRightarrow{*} x && \text{for some } f \in F && \text{[by construction]} \\ &\iff S \xRightarrow{*} x \text{ and } \hat{\delta}(q_0, x) = f && && \text{[by the lemma]} \\ &\iff x \in L(G) \text{ and } x \in L(M) && && \text{[by def. of the language of a CFG and DFA]} \\ &\iff x \in L_1 \cap L_2. \end{aligned}$$

□

C. Using the theorem

Returning to the language B from Equation (2), we can apply the theorem to prove that it's not regular. Specifically, consider the language C generated by the regular expression $\mathbf{a}^*\mathbf{b}^*\mathbf{c}^*$. The language C is regular because its given by a regular expression. Let's use it to prove that B is not context-free.

Proof. Assume B is context free. By the theorem, $B \cap C$ is context-free. However, $B \cap C = A$ where A is given in Equation (1) which we know is not context-free. This is a contradiction so B must not be context-free. $\square$

D. Bonus

I didn't do this in class, but let's apply the construction from the theorem to the following languages over the alphabet $\Sigma = \{\mathbf{a}, \mathbf{b}\}$.

$$D = \{x \mid x = x^R\}$$

$$E = \{x \mid |x| \text{ is even}\}.$$

Language D is context free and is given by the following grammar in CNF.

$$\begin{aligned} S &\rightarrow AX \mid BY \mid AA \mid BB \mid \mathbf{a} \mid \mathbf{b} \mid \varepsilon \\ T &\rightarrow AX \mid BY \mid AA \mid BB \mid \mathbf{a} \mid \mathbf{b} \\ X &\rightarrow TA \\ Y &\rightarrow TB \\ A &\rightarrow \mathbf{a} \\ B &\rightarrow \mathbf{b}. \end{aligned}$$

Language E is regular and given by the two-state DFA $M = (Q, \Sigma, \delta, s, F)$ where

$$\begin{aligned} Q &= \{e, o\} && \text{corresponding to even and odd length} \\ s &= e \\ F &= \{e\} \\ \delta(e, \mathbf{a}) &= \delta(e, \mathbf{b}) = o \\ \delta(o, \mathbf{a}) &= \delta(o, \mathbf{b}) = e. \end{aligned}$$

We start by adding rules $A_{q,r} \rightarrow B_{q,s}C_{s,r}$ corresponding to rules $A \rightarrow BC$ in P .

Rules from $S \rightarrow AX$

$$\begin{aligned} S_{e,e} &\rightarrow A_{e,e}X_{e,e} \mid A_{e,o}X_{o,e} \\ S_{e,o} &\rightarrow A_{e,e}X_{e,o} \mid A_{e,o}X_{o,o} \\ S_{o,e} &\rightarrow A_{o,e}X_{e,e} \mid A_{o,o}X_{o,e} \\ S_{o,o} &\rightarrow A_{o,e}X_{e,o} \mid A_{o,o}X_{o,o} \end{aligned}$$

Rules from $S \rightarrow BY$

$$\begin{aligned} S_{e,e} &\rightarrow B_{e,e}Y_{e,e} \mid B_{e,o}Y_{o,e} \\ S_{e,o} &\rightarrow B_{e,e}Y_{e,o} \mid B_{e,o}Y_{o,o} \\ S_{o,e} &\rightarrow B_{o,e}Y_{e,e} \mid B_{o,o}Y_{o,e} \\ S_{o,o} &\rightarrow B_{o,e}Y_{e,o} \mid B_{o,o}Y_{o,o} \end{aligned}$$

Rules from $S \rightarrow AA$

$$\begin{aligned} S_{e,e} &\rightarrow A_{e,e}A_{e,e} \mid A_{e,o}A_{o,e} \\ S_{e,o} &\rightarrow A_{e,e}A_{e,o} \mid A_{e,o}A_{o,o} \\ S_{o,e} &\rightarrow A_{o,e}A_{e,e} \mid A_{o,o}A_{o,e} \\ S_{o,o} &\rightarrow A_{o,e}A_{e,o} \mid A_{o,o}A_{o,o} \end{aligned}$$

Rules from $S \rightarrow BB$

$$\begin{aligned} S_{e,e} &\rightarrow B_{e,e}B_{e,e} \mid B_{e,o}B_{o,e} \\ S_{e,o} &\rightarrow B_{e,e}B_{e,o} \mid B_{e,o}B_{o,o} \\ S_{o,e} &\rightarrow B_{o,e}B_{e,e} \mid B_{o,o}B_{o,e} \\ S_{o,o} &\rightarrow B_{o,e}B_{e,o} \mid B_{o,o}B_{o,o} \end{aligned}$$

Rules from $T \rightarrow AX$

$$\begin{aligned} T_{e,e} &\rightarrow A_{e,e}X_{e,e} \mid A_{e,o}X_{o,e} \\ T_{e,o} &\rightarrow A_{e,e}X_{e,o} \mid A_{e,o}X_{o,o} \\ T_{o,e} &\rightarrow A_{o,e}X_{e,e} \mid A_{o,o}X_{o,e} \\ T_{o,o} &\rightarrow A_{o,e}X_{e,o} \mid A_{o,o}X_{o,o} \end{aligned}$$

Rules from $T \rightarrow BY$

$$\begin{aligned} T_{e,e} &\rightarrow B_{e,e}Y_{e,e} \mid B_{e,o}Y_{o,e} \\ T_{e,o} &\rightarrow B_{e,e}Y_{e,o} \mid B_{e,o}Y_{o,o} \\ T_{o,e} &\rightarrow B_{o,e}Y_{e,e} \mid B_{o,o}Y_{o,e} \\ T_{o,o} &\rightarrow B_{o,e}Y_{e,o} \mid B_{o,o}Y_{o,o} \end{aligned}$$

Rules from $T \rightarrow AA$

$$\begin{aligned} T_{e,e} &\rightarrow A_{e,e}A_{e,e} \mid A_{e,o}A_{o,e} \\ T_{e,o} &\rightarrow A_{e,e}A_{e,o} \mid A_{e,o}A_{o,o} \\ T_{o,e} &\rightarrow A_{o,e}A_{e,e} \mid A_{o,o}A_{o,e} \\ T_{o,o} &\rightarrow A_{o,e}A_{e,o} \mid A_{o,o}A_{o,o} \end{aligned}$$

Rules from $T \rightarrow BB$

$$\begin{aligned} T_{e,e} &\rightarrow B_{e,e}B_{e,e} \mid B_{e,o}B_{o,e} \\ T_{e,o} &\rightarrow B_{e,e}B_{e,o} \mid B_{e,o}B_{o,o} \\ T_{o,e} &\rightarrow B_{o,e}B_{e,e} \mid B_{o,o}B_{o,e} \\ T_{o,o} &\rightarrow B_{o,e}B_{e,o} \mid B_{o,o}B_{o,o} \end{aligned}$$

Rules from $X \rightarrow TA$

$$\begin{aligned} X_{e,e} &\rightarrow T_{e,e}A_{e,e} \mid T_{e,o}A_{o,e} \\ X_{e,o} &\rightarrow T_{e,e}A_{e,o} \mid T_{e,o}A_{o,o} \\ X_{o,e} &\rightarrow T_{o,e}A_{e,e} \mid T_{o,o}A_{o,e} \\ X_{o,o} &\rightarrow T_{o,e}A_{e,o} \mid T_{o,o}A_{o,o} \end{aligned}$$

Rules from $Y \rightarrow TB$

$$\begin{aligned} Y_{e,e} &\rightarrow T_{e,e}B_{e,e} \mid T_{e,o}B_{o,e} \\ Y_{e,o} &\rightarrow T_{e,e}B_{e,o} \mid T_{e,o}B_{o,o} \\ Y_{o,e} &\rightarrow T_{o,e}B_{e,e} \mid T_{o,o}B_{o,e} \\ Y_{o,o} &\rightarrow T_{o,e}B_{e,o} \mid T_{o,o}B_{o,o} \end{aligned}$$

Next we add the rules $A_{q,r} \rightarrow t$ corresponding to rules $A \rightarrow t$ such that $\delta(q, t) = r$.

Rules from $S \rightarrow \mathbf{a}$

$$\begin{aligned} S_{e,o} &\rightarrow \mathbf{a} \\ S_{o,e} &\rightarrow \mathbf{a} \end{aligned}$$

Rules from $S \rightarrow \mathbf{b}$

$$\begin{aligned} S_{e,o} &\rightarrow \mathbf{b} \\ S_{o,e} &\rightarrow \mathbf{b} \end{aligned}$$

Rules from $T \rightarrow \mathbf{a}$

$$\begin{aligned} T_{e,o} &\rightarrow \mathbf{a} \\ T_{o,e} &\rightarrow \mathbf{a} \end{aligned}$$

Rules from $T \rightarrow \mathbf{b}$

$$\begin{aligned} T_{e,o} &\rightarrow \mathbf{b} \\ T_{o,e} &\rightarrow \mathbf{b} \end{aligned}$$

Rules from $A \rightarrow \mathbf{a}$

$$\begin{aligned} A_{e,o} &\rightarrow \mathbf{a} \\ A_{o,e} &\rightarrow \mathbf{a} \end{aligned}$$

Rules from $B \rightarrow \mathbf{b}$

$$\begin{aligned} B_{e,o} &\rightarrow \mathbf{b} \\ B_{o,e} &\rightarrow \mathbf{b} \end{aligned}$$

Finally, we need to add rules for the start nonterminal S' . There's exactly 1 element in F , namely e . And ε is an element of both D and E so we need a rule for that as well.

$$S' \rightarrow S_{e,e} \mid \varepsilon$$

That's a lot of rules!

For comparison, here's a hand-crafted CFG for the language $D \cap E$.

$$S \rightarrow \mathbf{aSa} \mid \mathbf{bSb} \mid \varepsilon$$

Fear not. I will not ask you to perform this construction by hand.