بسم اللّه الرّحمن الرّحیم

دانشگاه صنعتی اصفهان ــ دانشکدهٔ مهندسی برق و کامپیوتر
(نیم‌سال تحصیلی ۴۰۲۱)

نظریهٔ زبان‌ها و ماشین‌ها

حسین فلسفین

چند مثال دیگر

Example 1: ***Give a CFG generating the following language: The complement of the language*** $A = \{a^n b^n | n \geq 0\}$***.***

Solution:

$$\overline{A} = \{a^m b^n | m \neq n\} \cup L\left((a \cup b)^* b (a \cup b)^* a (a \cup b)^*\right)$$

$$\begin{aligned}
S &\rightarrow S_1 | S_2 | S_3 \\
S_1 &\rightarrow aS_1b | aS_1 | a \\
S_2 &\rightarrow aS_2b | S_2b | b \\
S_3 &\rightarrow XbXaX \\
X &\rightarrow aX | bX | \varepsilon
\end{aligned}$$

Example 2: ***Give a CFG generating the following language:***

$$\{a^i b^j c^k | j \neq i+k\} \subseteq L(a^* b^* c^*) \subseteq \{a,b,c\}^*.$$

Solution:

$$L_1 \cup L_2 = \{a^i b^j c^k | j > i+k\} \cup \{a^i b^j c^k | j < i+k\}$$

$$S \rightarrow S_1 | S_2$$

$$L_1 = MNP = \{a^i b^i | i \geq 0\} \circ \{b^m | m > 0\} \circ \{b^k c^k | k \geq 0\}$$

$$S_1 \rightarrow S_M S_N S_P,\ S_M \rightarrow aS_M b|\varepsilon,\ S_N \rightarrow bS_N|b,\ S_P \rightarrow bS_P c|\varepsilon$$

$$L_2 = L_3 \cup L_4 = \{a^i b^j c^k | j < i\} \cup \{a^i b^j c^k | i \leq j < i + k\}$$

$$S_2 \rightarrow S_3 | S_4$$

$$L_3 = QRT = \{a^i | i > 0\} \circ \{a^i b^i | i \geq 0\} \circ \{c^i | i \geq 0\}$$

$$S_3 \rightarrow S_Q S_R S_T,\ S_Q \rightarrow a S_Q | a,\ S_R \rightarrow a S_R b | \varepsilon,\ S_T \rightarrow c S_T | \varepsilon$$

$$L_4 = UVW = \{a^i b^i | i \geq 0\} \circ \{b^i c^i | i \geq 0\} \circ \{c^i | i > 0\}$$

$$S_4 \rightarrow S_U S_V S_W,\ S_U \rightarrow a S_U b | \varepsilon,\ S_V \rightarrow b S_V c | \varepsilon,\ S_W \rightarrow c S_W | c$$

Example 3: ***Give a CFG generating the following language:***

$$\{w\#x | w^R \textbf{ is a substring of } x \textbf{ for } w, x \in \{0,1\}^*\}.$$

Solution:

$$\begin{aligned} S &\rightarrow TX \\ T &\rightarrow 0T0|1T1|\#X \\ X &\rightarrow 0X|1X|\varepsilon \end{aligned}$$

Example 4: *Here is a somewhat harder example. Let us create a CFG to generate the nonpalindromes over* $\{a, b\}$.

Solution:

$$\begin{aligned} S &\rightarrow aSa|bSb|aTb|bTa \\ T &\rightarrow aTa|aTb|bTa|bTb|\varepsilon|a|b \end{aligned}$$

The basic idea is that if a string is a nonpalindrome, then there must be at least one position such that the character in that position does not match the character in the corresponding position from the end. The productions $S \rightarrow aSa$ *and* $S \rightarrow bSb$ *are used to generate a prefix and suffix that match properly, but eventually one of the two productions involving* T *on the right-hand side must be used, at which point a mismatch is introduced. Now the remaining symbols can either match or not match, which accounts for the remaining productions involving* T.

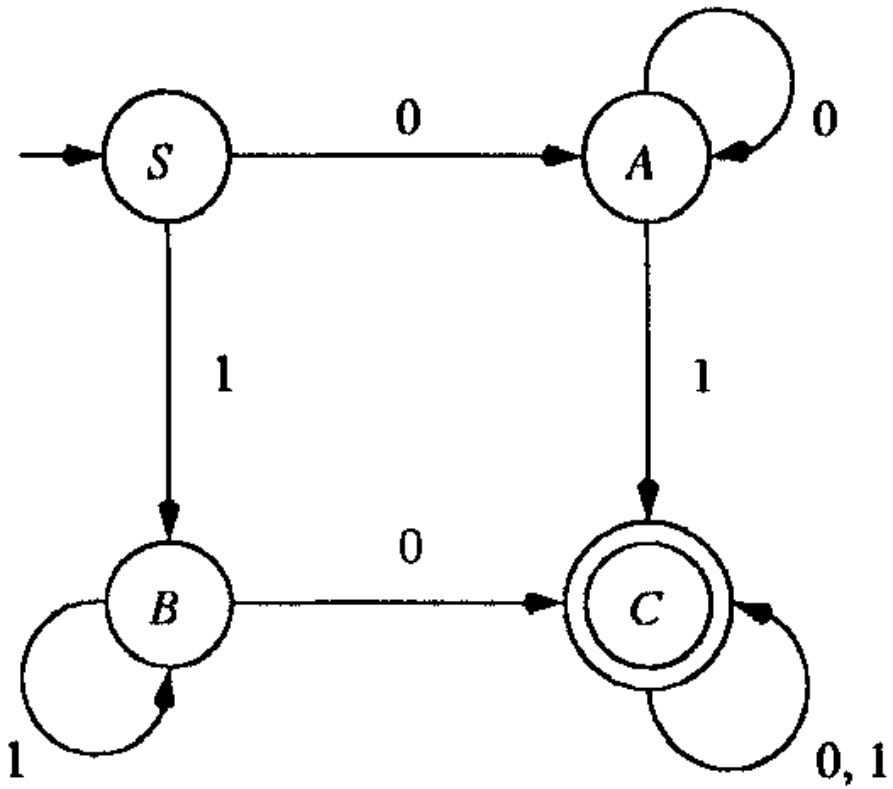
Another CFG for the same language:

$$
\begin{aligned}
R &\rightarrow XRX \mid S \\
S &\rightarrow \mathtt{a}T\mathtt{b} \mid \mathtt{b}T\mathtt{a} \\
T &\rightarrow XTX \mid X \mid \varepsilon \\
X &\rightarrow \mathtt{a} \mid \mathtt{b}
\end{aligned}
$$

Every regular language is also a context-free language

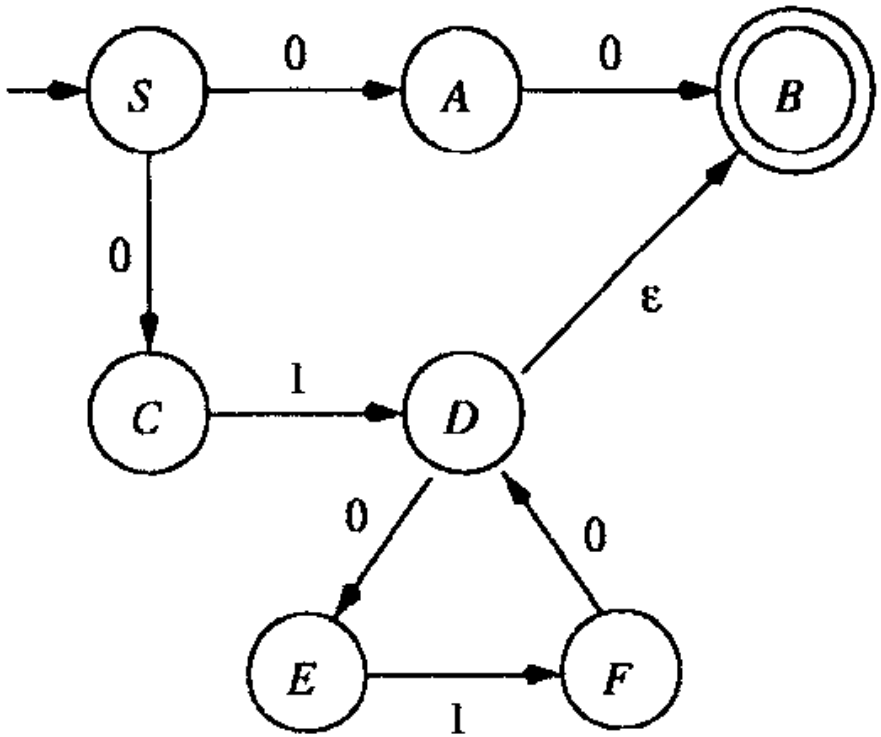
For each regular language L, there exists a context-free grammar G such that $L = L(G)$. Constructing a CFG for a language that happens to be regular is easy if you can first construct a DFA for that language. You can convert any DFA into an equivalent CFG as follows. Make a variable R_i for each state q_i of the DFA. Add the rule $R_i \to aR_j$ to the CFG if $\delta(q_i, a) = q_j$ is a transition in the DFA. Add the rule $R_i \to \varepsilon$ if q_i is an accept state of the DFA. Make R_0 the start variable of the grammar, where q_0 is the start state of the machine. Verify on your own that the resulting CFG generates the same language that the DFA recognizes.

Example:



$$S \to 0A|1B,\ A \to 0A|1C,\ B \to 0C|1B,\ C \to 0C|1C|\varepsilon$$

Example:



$$S \longrightarrow 0A \mid 0C, \qquad A \longrightarrow 0B, \qquad B \longrightarrow \varepsilon,$$
$$C \longrightarrow 1D, \qquad D \longrightarrow 0E \mid B, \qquad E \longrightarrow 1F,$$
$$F \longrightarrow 0D.$$

خواص بستاری زبان‌های مستقل از متن

Theorem: The family of CFLs is closed under union, concatenation, and star-closure.

Proof: *Let L_1 and L_2 be two context-free languages generated by the context-free grammars $G_1 = (V_1, \Sigma_1, R_1, S_1)$ and $G_2 = (V_2, \Sigma_2, R_2, S_2)$, respectively. We can assume without loss of generality that the sets V_1 and V_2 are disjoint. Consider now the language $L(G_3)$, generated by the grammar*

$$G_3 = (V_1 \cup V_2 \cup \{S_3\}, \Sigma_1 \cup \Sigma_2, R_3, S_3),$$

where S_3 is a variable not in $V_1 \cup V_2$. The productions of G_3 are all the productions of G_1 and G_2, together with an alternative starting production that allows us to use one or the other grammars. More precisely, $R_3 = R_1 \cup R_2 \cup \{S_3 \to S_1 | S_2\}$. Obviously, G_3 is a context-free grammar, so that $L(G_3)$ is a context-free language. But it is easy to see that $L(G_3) = L_1 \cup L_2$. Suppose, for instance, that

$w \in L_1$. *Then* $S_3 \Rightarrow S_1 \Rightarrow^* w$ *is a possible derivation in grammar* G_3. *A similar argument can be made for* $w \in L_2$. *Also, if* $w \in L(G_3)$, *then either* $S_3 \Rightarrow S_1$ *or* $S_3 \Rightarrow S_2$ *must be the first step of the derivation. Suppose* $S_3 \Rightarrow S_1$ *is used. Since sentential forms derived from* S_1 *have variables in* V_1, *and* V_1 *and* V_2 *are disjoint, the derivation can involve productions in* R_1 *only. Hence* w *must be in* L_1. *Alternatively, if* $S_3 \Rightarrow S_2$ *is used first, then* w *must be in* L_2 *and it follows that* $L(G_3)$ *is the union of* L_1 *and* L_2.

Next, *consider* $G_4 = (V_1 \cup V_2 \cup \{S_4\}, \Sigma_1 \cup \Sigma_2, R_4, S_4)$. *Here again* S_4 *is a new variable and* $R_4 = R_1 \cup R_2 \cup \{S_4 \rightarrow S_1 S_2\}$. *Then* $L(G_4) = L(G_1) \circ L(G_2)$ *follows easily.*

Finally, *consider* $L(G_5)$ *with* $G_5 = (V_1 \cup \{S_5\}, \Sigma_1, R_5, S_5)$, *where* S_5 *is a new variable and* $R_5 = R_1 \cup \{S_5 \rightarrow S_1 S_5 | \varepsilon\}$. *Then* $L(G_5) = L(G_1)^*$. *Thus we have shown that the family of context-free languages is closed under union, concatenation, and star-closure.*

دربارهٔ دو عملگر اشتراک و مکمل‌گیری چه می‌توان گفت؟

The family of context-free languages is not closed under intersection and complementation.

Proof: *Consider the two languages* $L_1 = \{a^n b^n c^m : n \geq 0, m \geq 0\}$ *and* $L_2 = \{a^n b^m c^m : n \geq 0, m \geq 0\}$. *There are several ways one can show that* L_1 *and* L_2 *are context free. For instance, a grammar for* L_1 *is*

$$\begin{aligned} S &\rightarrow S_1 S_2, \\ S_1 &\rightarrow aS_1 b | \varepsilon, \\ S_2 &\rightarrow cS_2 | \varepsilon. \end{aligned}$$

Alternatively, we note that L_1 *is the concatenation of two context-free languages, so it is context free. But it can be shown that* $L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\}$ *is not context free. Thus, the family of*

context-free languages is not closed under intersection. The second part of the theorem follows from the set identity

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

If the family of context-free languages were closed under complementation, then the right side of the above expression would be a context-free language for any context-free L_1 and L_2. But this contradicts what we have just shown, that the intersection of two context-free languages is not necessarily context-free. Consequently, the family of context-free languages is not closed under complementation.

Example (A CFL Whose Complement Is Not a CFL): ***Surprisingly, although we can show that the language*** $L_1 = \{ww|w \in \{a,b\}^*\}$ ***is not a CFL, its complement is. Let*** L ***be the complement*** $\{a,b\}^* \setminus L_1$. L ***contains every string in*** $\{a,b\}^*$ ***of odd length. If*** $x \in L$ ***and*** $|x| = 2n$ ***for some*** $n \geq 1$, ***then for some*** k ***with*** $1 \leq k \leq n$, ***the*** k***th and*** $(n+k)$***th symbols of*** x ***are different (say*** a ***and*** b, ***respectively). There are*** $k-1$ ***symbols before the*** a, $n-1$ ***symbols between the two, and*** $n-k$ ***symbols after the*** b. ***But instead of thinking of the*** $n-1$ ***symbols between the two as*** $n-k$ ***and then*** $k-1$ ***(the remaining symbols in the first half, followed by the symbols in the second half that precede the*** b***),***

we can think of them as $k-1$ ***and then*** $n-k$.

In other words, x is the concatenation of two odd-length strings, the first with a in the middle and $k-1$ symbols on either side, and the second with b in the middle and $n-k$ symbols on either side. Conversely, every concatenation of two such odd-length strings is in L. The conclusion is that L can be generated by the CFG with productions

$$\begin{aligned} S &\rightarrow A|B|AB|BA, \\ A &\rightarrow EAE|a, \\ B &\rightarrow EBE|b \\ E &\rightarrow a|b. \end{aligned}$$

The variables A and B generate odd-length strings with middle symbol a and b, respectively, and together generate all odd-length strings. Therefore, L is a CFL whose complement is not a CFL.

نکتهٔ بسیار مهم دربارهٔ اشتراک یک زبان منظم و یک زبان مستقل از متن

While the intersection of two context-free languages may produce a language that is not context free, the closure property holds if one of the languages is regular.

بعد از معرفی ***PDA***ها، این قضیه را اثبات خواهیم کرد.

تمرین (نیاز به تحویل ندارد): یک رشته باینری با طول ۸ و یک رشته باینری با طول ۱۰ مثال بزنید که در زبان $L_1 = \{ww|w \in \{a,b\}^*\}$ قرار **ندارند**. سپس، درایویشن نظیر هریک از این دو رشته را باتوجه به گرامر معرفی‌شده در اسلاید شمارهٔ ۱۶ بنویسید.