Properties of CFLs

Theorem 1 *CFLs are closed under union, concatenation, and Kleene Star.*

Proof

- $G_1 = (V_1, \Sigma_1, R_1, S_1)$, $G_2 = (V_2, \Sigma_2, R_2, S_2)$.
- Let

  $$\begin{aligned} G = (& V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, \\ & R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}, S). \end{aligned}$$

  Then

  $$L(G) = L(G_1) \cup L(G_2).$$

- Let $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{S \to S_1 S_2\}, S)$. Then

  $$L(G) = L(G_1)L(G_2).$$

- Let $G = (V_1, \Sigma_1, R_1 \cup \{S_1 \to e, S_1 \to S_1 S_1\}, S_1)$. Then

  $$L(G) = L(G_1)^*.$$

But,intersection of two CFLs **might not be** a CFL!

Theorem 2 *The intersection of a CFL and a regular language is a CFL.*

Proof:

- Given:
  - $L_1 = L(M_1)$ for some PA $M_1 = (K_1, \Sigma_1, \Gamma_1, \Delta_1, s_1, F_1)$.
  - $L_2 = L(M_2)$ for some DFA $M_2 = (K_2, \Sigma_2, \delta_2, s_2, F_2)$.
- Need to show: $L_1 \cap L_2 = L(M)$ for some PA, $M = (K, \Sigma, \Gamma, \Delta, s, F)$.
- The idea: Construct a PA M that operates in the same way as M_1 except that it also keeps the track of the change in state in M_2 caused by reading the same input.
- The construction: $K = K_1 \times K_2$, $\Sigma = \Sigma_1 \cup \Sigma_2$, $\Gamma = \Gamma_1$, $s = (s_1, s_2)$, $F = F_1 \times F_2$;
- And

  $$(((q_1, q_2), u, \beta), ((p_1, p_2), \gamma) \in \Delta$$

  iff

  $$((q_1, u, \beta), (p_1, \gamma) \in \Delta_1,$$

  and

  $$(q_2, u) \vdash^*_{M_2} (p_2, e).$$

The pumping theorem

Theorem 3 *Given a CFG G. There is a number K such that for any $w \in L(G)$, if $|w| > K$ then w can be written as*

$$w = uvxyz$$

such that either $|v| > 0$ or $|y| > 0$ and for any $n \geq 0$

$$uv^n xy^n x \in L(G).$$

Proof:

- *Branching factor* of a parse tree: the maximum number of children a node can have.
- The branching factors of the parse trees of G is upper bounded by the maximum number p of symbols on the right hand side of the rules.
- Let m be the number of nonterminals and let $K = p^m$.
- For any $w \in L(G)$, let w be the yield of parse tree T.
- If $|w| > K$, then there must be a path (from root to a leave) that is of length $> m$ and hence contains two appearances of a nonterminal A.

Figure 3-6 here.

- $w = uvxyz$ and either $|v| > 0$ or $|y| > 0$.
- $uv^n xy^n z \in L(G)$ because we can repeat the A-A subtree n times.

Properties of CFLs

The pumping theorem implies that

Theorem 4 $L = \{a^n b^n c^n : n \geq 0\}$ *is not context-free.*

Proof: See textbook.

- $L_1 = \{a^n b^n c^m : m, n \geq 0\}$ is a CFL.

$$S \to TC, T \to aTb, T \to e, C \to cC, C \to e$$

- $L_2 = \{a^m b^n c^n : m, n \geq 0\}$ is a CFL.

$$S \to AT, T \to bTc, T \to e, A \to aA, A \to e$$

- But $L = L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\}$ is not a CFL.

Therefore

Theorem 5 *The CFLs are not closed under intersection. Nor under complementation.*

Proof for the second part: If CFLs were closed under complementation, then because $L_1 \cap L_2 = \overline{\overline{L1} \cap \overline{L2}}$ they must be also be closed under intersection. A contradiction. $\square$

Algorithmic properties

Theorem 6 *There are algorithms for answering the following questions about CFLs:*

1. *Given G and w, $w \in L(G)$?*
2. *$L(G) = \emptyset$?*

Note*: No algorithm can answer this question "does $L(G_1) = L(G_2)$" .*

Proof: Proof of (1).

- If every rule $A \to u$ is such that u is either a terminal or $|u| > 2$,
- then in every parse tree, every internal node has at least two children.
- Hence any parse tree with height $> |w|$ has a yield with length $> |w|$; the yield cannot be w.
- Therefore we can determine if $w \in L(G)$ by inspecting all parse trees with height $\leq |w|$.

- What if G contains rules $A \to u$ where u is e or another nonterminal? Let's call such a rule a *problem rule*.
- Answer: eliminate them as follows.
  1. For any problem rule $A \to u$ and any rule of the form $B \to xAy$,

     Add rule $B \to xuy$.
  2. Repeat the process for new problem rules produced until no new rules can be added.
  3. Eliminate all the problem rules except $S \to e$, if it is present.
- The elimination process does not change the language generated:
  1. Every new rule is a combination of two or more original rules.
  2. When no new rules can be added, any application of the problem rules in derivation can be replaced by some of the new rules.
- $S \to e$, if present, is needed only when $w = e$.
- For any $w \neq e$, $w \in L(G)$ iff it can be derived without using $S \to e$.

- The arguments on the previous slide apply. $\square$

Proof of (2):

- Let K be the number in the pumping theorem.
- If $L(G) \neq \emptyset$, then there is one $w \in L(G)$ that is of length $\leq K$.
- There are at most Σ^K strings with length $\leq K$.
- We check if each of them is in $L(G)$, which is possible due to item (1).
- Consequently, we can tell if $L(G) = \emptyset$.