

1 Closure Properties

1.1 Intersection and Complementation

Intersection

Proposition 1. *CFLs are not closed under intersection*

Proof. • $L_1 = \{a^i b^j c^j \mid i, j \geq 0\}$ is a CFL

– Generated by a grammar with rules $S \rightarrow XY$; $X \rightarrow aXb \mid \epsilon$; $Y \rightarrow cY \mid \epsilon$.

• $L_2 = \{a^i b^j c^j \mid i, j \geq 0\}$ is a CFL.

– Generated by a grammar with rules $S \rightarrow XY$; $X \rightarrow aX \mid \epsilon$; $Y \rightarrow bYc \mid \epsilon$.

• But $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$, which we will see soon, is not a CFL. □

Intersection with Regular Languages

Proposition 2. *If L is a CFL and R is a regular language then $L \cap R$ is a CFL.*

Proof. Let P be the PDA that accepts L , and let M be the DFA that accepts R . A new PDA P' will simulate P and M simultaneously on the same input and accept if both accept. Then P' accepts $L \cap R$.

- The stack of P' is the stack of P
- The state of P' at any time is the pair (state of P , state of M)
- These determine the transition function of P'
- The final states of P' are those in which both the state of P and state of M are accepting.

More formally, let $M = (Q_1, \Sigma, \delta_1, q_1, F_1)$ be a DFA such that $\mathbf{L}(M) = R$, and $P = (Q_2, \Sigma, \Gamma, \delta_2, q_2, F_2)$ be a PDA such that $\mathbf{L}(P) = L$. Then consider $P' = (Q, \Sigma, \Gamma, \delta, q_0, F)$ such that

- $Q = Q_1 \times Q_2$
- $q_0 = (q_1, q_2)$
- $F = F_1 \times F_2$

$$\delta((p, q), x, a) = \begin{cases} \{(p, q'), b \mid (q', b) \in \delta_2(q, x, a)\} & \text{when } x = \epsilon \\ \{(p', q'), b \mid p' = \delta_1(p, x) \text{ and } (q', b) \in \delta_2(q, x, a)\} & \text{when } x \neq \epsilon \end{cases}$$

One can show by induction on the number of computation steps, that for any $w \in \Sigma^*$

$$\langle q_0, \epsilon \rangle \xrightarrow{w}_{P'} \langle (p, q), \sigma \rangle \text{ iff } q_1 \xrightarrow{w}_M p \text{ and } \langle q_2, \epsilon \rangle \xrightarrow{w}_P \langle q, \sigma \rangle$$

The proof of this statement is left as an exercise. Now as a consequence, we have $w \in L(P')$ iff $\langle q_0, \epsilon \rangle \xrightarrow{w}_{P'} \langle (p, q), \sigma \rangle$ such that $(p, q) \in F$ (by definition of PDA acceptance) iff $\langle q_0, \epsilon \rangle \xrightarrow{w}_{P'} \langle (p, q), \sigma \rangle$ such that $p \in F_1$ and $q \in F_2$ (by definition of F) iff $q_1 \xrightarrow{w}_M p$ and $\langle q_2, \epsilon \rangle \xrightarrow{w}_P \langle q, \sigma \rangle$ and $p \in F_1$ and $q \in F_2$ (by the statement to be proved as exercise) iff $w \in L(M)$ and $w \in L(P)$ (by definition of DFA acceptance and PDA acceptance). \square

Why does this construction not work for intersection of two CFLs?

Complementation

Proposition 3. *Context-free languages are not closed under complementation.*

Proof. [**Proof 1**] Suppose CFLs were closed under complementation. Then for any two CFLs L_1, L_2 , we have

- $\overline{L_1}$ and $\overline{L_2}$ are CFL. Then, since CFLs closed under union, $\overline{L_1} \cup \overline{L_2}$ is CFL. Then, again by hypothesis, $\overline{\overline{L_1} \cup \overline{L_2}}$ is CFL.
- i.e., $L_1 \cap L_2$ is a CFL

i.e., CFLs are closed under intersection. Contradiction!

[**Proof 2**] $L = \{x \mid x \text{ not of the form } ww\}$ is a CFL.

- L generated by a grammar with rules $X \rightarrow a|b, A \rightarrow a|XAX, B \rightarrow b|XBX, S \rightarrow A|B|AB|BA$

But $\overline{L} = \{ww \mid w \in \{a, b\}^*\}$ we will see is not a CFL! \square

Set Difference

Proposition 4. *If L_1 is a CFL and L_2 is a CFL then $L_1 \setminus L_2$ is not necessarily a CFL*

Proof. Because CFLs not closed under complementation, and complementation is a special case of set difference. (How?) \square

Proposition 5. *If L is a CFL and R is a regular language then $L \setminus R$ is a CFL*

Proof. $L \setminus R = L \cap \overline{R}$ \square

1.2 Homomorphisms

Homomorphism

Proposition 6. *Context free languages are closed under homomorphisms.*

Proof. Let $G = (V, \Sigma, R, S)$ be the grammar generating L , and let $h : \Sigma^* \rightarrow \Gamma^*$ be a homomorphism. A grammar $G' = (V', \Gamma, R', S')$ for generating $h(L)$:

- Include all variables from G (i.e., $V' \supseteq V$), and let $S' = S$
- Treat terminals in G as variables. i.e., for every $a \in \Sigma$
 - Add a new variable X_a to V'
 - In each rule of G , if a appears in the RHS, replace it by X_a
- For each X_a , add the rule $X_a \rightarrow h(a)$

G' generates $h(L)$. (*Exercise!*)

□

Example 7. Let G have the rules $S \rightarrow 0S0 | 1S1 | \epsilon$.

Consider the homomorphism $h : \{0, 1\}^* \rightarrow \{a, b\}^*$ given by $h(0) = aba$ and $h(1) = bb$.

Rules of G' s.t. $\mathbf{L}(G') = \mathbf{L}(h(L))$:

$$\begin{aligned} S &\rightarrow X_0 S X_0 | X_1 S X_1 | \epsilon \\ X_0 &\rightarrow aba \\ X_1 &\rightarrow bb \end{aligned}$$

1.3 Inverse Homomorphisms

Inverse Homomorphisms

Recall: For a homomorphism h , $h^{-1}(L) = \{w \mid h(w) \in L\}$

Proposition 8. *If L is a CFL then $h^{-1}(L)$ is a CFL*

Proof Idea

For regular language L : the DFA for $h^{-1}(L)$ on reading a symbol a , simulated the DFA for L on $h(a)$. Can we do the same with PDAs?

- Key idea: store $h(a)$ in a “buffer” and process symbols from $h(a)$ one at a time (according to the transition function of the original PDA), and the next input symbol is processed only after the “buffer” has been emptied.
- Where to store this “buffer”? In the state of the new PDA!

Proof. Let $P = (Q, \Delta, \Gamma, \delta, q_0, F)$ be a PDA such that $\mathbf{L}(P) = L$. Let $h : \Sigma^* \rightarrow \Delta^*$ be a homomorphism such that $n = \max_{a \in \Sigma} |h(a)|$, i.e., every symbol of Σ is mapped to a string under h of length at most n . Consider the PDA $P' = (Q', \Sigma, \Gamma, \delta', q'_0, F')$ where

- $Q' = Q \times \Delta^{\leq n}$, where $\Delta^{\leq n}$ is the collection of all strings of length at most n over Δ .
- $q'_0 = (q_0, \epsilon)$
- $F' = F \times \{\epsilon\}$
- δ' is given by

$$\delta'((q, v), x, a) = \begin{cases} \{(q, h(x)), \epsilon\} & \text{if } v = a = \epsilon \\ \{(p, u), b \mid (p, b) \in \delta(q, y, a)\} & \text{if } v = yu, x = \epsilon, \text{ and } y \in \Delta \end{cases}$$

and $\delta'(\cdot) = \emptyset$ in all other cases.

We can show by induction that for every $w \in \Sigma^*$

$$\langle q'_0, \epsilon \rangle \xrightarrow{w}_{P'} \langle (q, v), \sigma \rangle \text{ iff } \langle q_0, \epsilon \rangle \xrightarrow{w'}_P \langle q, \sigma \rangle$$

where $h(w) = w'v$. Again this induction proof is left as an exercise. Now, $w \in \mathbf{L}(P')$ iff $\langle q'_0, \epsilon \rangle \xrightarrow{w}_{P'} \langle (q, v), \sigma \rangle$ where $q \in F$ (by definition of PDA acceptance and F') iff $\langle q_0, \epsilon \rangle \xrightarrow{h(w)}_P \langle q, \sigma \rangle$ (by exercise) iff $h(w) \in \mathbf{L}(P)$ (by definition of PDA acceptance). Thus, $\mathbf{L}(P') = h^{-1}(\mathbf{L}(P)) = h^{-1}(L)$. \square
