Theory of Computation

PSet-3

06 October 2025

This is a compilation of solutions for PSet-3 of Theory of Computation class instructed by Prof. C Aishwariya. The solutions are of original design.

Table of contents

§0. Problems

§1. Problem 1

Exercise

Let $\Sigma = \{a, b, \#\}$. Consider the language

$$L = \{(w\#)^* \mid w \in \{a, b\}^*\}$$

. That is, L consists of all strings that can be written as zero or more repetitions of a word $w \in \{a, b\}^*$ followed by the delimiter symbol $\#$.

(a) is L context free? Prove.

(b) Prove that the complement language $\overline{L}$ is context-free by constructing a pushdown automaton (PDA) that recognizes $\overline{L}$.

(a)

Proof. We will prove this by game With demon (Kozen, equivalent to Pumping lemma).

For a given natural $k > 0$, we pick $z = a^k b^k \# a^k b^k \#$. We claim that for any choice of $uvwxy$ such that $|vx| > 0$ and $|vwx| \leq k$.

- **Case 1** If vwx lie before the first $\#$ then for $i = 2$, $uw^i vx^i y \notin L$ as the words corresponding to both the hash's are different as both v, x can't be ε and we didn't change the second word.
- **Case 2** If vwx lies after the first $\#$ then for $i = 2$, $uv^i wx^i v \notin L$ as:
  - **Case 2.1** If x doesn't contain $\#$ then $uv^i wx^i v \notin L$ as the words corresponding to both the hash's are different as both v, x can't be ε and we didn't change the second word.
  - **Case 2.2** If x contains $\#$, then $x = a^k b^k \#$ can only maintain the membership in L (as we can't change the number of word before the first hash) but then $|vwx| = 2k + 1 > k$ which is not possible. Thus, $uw^i vx^i v \notin L$
- **Case 3** If vwx has the first $\#$ then $v \in \bigcup_{i \leq p}(b^i \cup b^i \#)$ and $x \in \bigcup_{i \leq p}(a^i \cup \# a^i)$.

  But then, for $i = 2$, $uv^2 wx^2 y \notin L$ as the number of $b's$ in the word corresponding to the first hash is greater than the number of b's in the word corresponding to the second hash, the number of a's in the word corresponding to the first hash is lesser than the number of b's in the word corresponding to the second hash or both. This is as both v, x can't be ε.

  Thus, L is not context-free.

□

(b) Notice,

$$\overline{L} = \{wx|\ w \in \{a, b, \#\}^*, x \in \{a, b\}\} \cup \bigcup_{n}^{\infty} \{w_1 \# w_2 \# ... w_n \# \mid \exists i \text{ s.t } w_1 \neq w_i\}$$

$\{wx|\ w \in \{a, b, \#\}^*, x \in \{a, b\}\}$ is regular as it is recognized by the NFA $(Q, \Sigma, \Delta, S, F)$ where

- $Q = \{0, 1\}$
- $\Sigma = \{a, b, \#\}$
- $\Delta(0, a) = \{0, 1\}; \Delta(0, b) = \{0, 1\}; \Delta(0, \#) = \{0\}; \Delta(1, x) = \emptyset$
- $S = \{0\}$

- $F = \{1\}$

where we can only reach the accepting states if we end with a, b and not otherwise.

$\bigcup_n^\infty \{w_1 \# w_2 \# ... w_n \# \mid \exists i \text{ s.t } w_1 \neq w_i, w_1 \neq \varepsilon\}$ is a CFG as it is recognized by the PDA $(Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ where

- $Q = \{0, 1_a, 1_b, 2_a, 3_a, 4_a, 5_a, 2_b, 3_b, 4_b, 5_b\}$
- $\Sigma = \{a, b, \#\}$
- $\Gamma = \{a, b, x \perp\}$
- $q_0 = 0$
- $Z = \perp$
- $F = \{5_a, 5_b\}$

We have the following transitions

$$\Delta(0, \varepsilon, \perp) = \{(1_a, a), (1_b, b)\}$$

$$\Delta(1_a, a, \Gamma) = \{(1_a, x\Gamma), (2_a, \Gamma)\}$$
$$\Delta(1_a, b, \Gamma) = \{(1_a, x\Gamma)\}$$
$$\Delta(2_a, \#, \Gamma) = \{(3_a, \Gamma), (4_a, \Gamma)\}$$
$$\Delta(2_a, \{a, b\}, \Gamma) = \{(2_a, \Gamma)\}$$
$$\Delta(3_a, \{a, b\}, \Gamma) = \{(3_a, \Gamma)\}$$
$$\Delta(4_a, \#, \Gamma) = \{(3_a, \Gamma), (4_a, \Gamma)\}$$
$$\Delta(4_a, \{a, b\}, x) = \{(4_a, \varepsilon)\}$$
$$\Delta(4_a, b, a) = \{(5_a, \varepsilon)\}$$
$$\Delta(4_a, \#, x) = \{(5_a, \varepsilon)\}$$
$$\Delta(5_a, \Sigma, \perp) = \{5_a, \perp\}$$

$$\Delta(1_b, b, \Gamma) = \{(1_b, x\Gamma), (2_b, \Gamma)\}$$
$$\Delta(1_b, a, \Gamma) = \{(1_b, x\Gamma)\}$$
$$\Delta(2_b, \#, \Gamma) = \{(3_b, \Gamma), (4_b, \Gamma)\}$$
$$\Delta(2_b, \{a, b\}, \Gamma) = \{(2_b, \Gamma)\}$$
$$\Delta(3_b, \{a, b\}, \Gamma) = \{(3_b, \Gamma)\}$$
$$\Delta(4_b, \#, \Gamma) = \{(3_b, \Gamma), (4_b, \Gamma)\}$$
$$\Delta(4_b, \{a, b\}, x) = \{(4_b, \varepsilon)\}$$
$$\Delta(4_b, a, a) = \{(5_b, \varepsilon)\}$$
$$\Delta(4_b, \#, x) = \{(5_b, \varepsilon)\}$$
$$\Delta(5_b, \Sigma, \perp) = \{5_b, \perp\}$$

We can also consider a case where $w_1 = \varepsilon$ which is solved by a DFA accepting $\#^+\{a, b\}\{a, b, \#\}^*$ which is easy to construct.

We can combine these all with the NFA by adding a three new states H_1, H_2, H_3, H_4 with H_3, H_4 being accepting and transitions

$$\Delta(0, \#, \perp) = \{(H_1, \perp), (H_2, \perp)\}$$
$$\Delta(0, \{a, b\}, \perp) = \{(H_2, \perp)\}$$
$$\Delta(H_1, \#, \perp) = \{(H_1, \perp)\}$$
$$\Delta(H_1, \{a, b\}, \perp) = \{(H_4, \perp)\}$$
$$\Delta(H_4, \Sigma, \perp) = \{(H_4, \perp)\}$$
$$\Delta(H_2, \#, \perp) = \{(H_2, \perp)\}$$
$$\Delta(H_2, \{a, b\}, \perp) = \{(H_2, \perp), (H_3, \perp)\}$$

§2. Problem 2

Exercise

Given a language L on the finite alphabet Σ, we define the following operators

$$\text{cyclic}(L) = \{yx \mid x, y \in \Sigma^* \text{ s.t. } xy \in L\}$$

which are all words that are formed by cyclic-shifting letters of some word in L

$$\text{permutation}(L) = \{y \mid \exists x \in L \text{ s.t. } \forall \sigma \in \Sigma, \quad |x|_\sigma = |y|_\sigma\}$$

which are all words that are formed by permuting the letters of some word in L.

(a) Given a regular language R, is $\text{cyclic}(R)$:

i. regular, but not context-free

ii. context-free

iii. not context free

(b) Given a context-free language G, is $\text{cyclic}(G)$:

i. regular, but not context-free

ii. context-free

iii. not context free

(c) Given a regular language R, is $\text{permutation}(R)$:

i. regular, but not context-free

ii. context-free

iii. not context free

(d) Given a context-free language G, is $\text{permutation}(G)$:

i. regular, but not context-free

ii. context-free

iii. not context free

(a) If language R has DFA $(Q, \Sigma, \delta, q_0, F)$ then we claim that a NFA $(Q', \Sigma, \Delta, Q', F')$ recognizes $\text{cyclic}(R)$.

We define $f_w = \hat{\delta}(q, w)$ for $w \in \Sigma^*$.

- $Q' = \left\{f_{w_{\text{left}}} \mid f_w \in Q^Q\right\} \cup \left\{f_{w_{\text{right}}} \mid f_w \in Q^Q\right\}$
- $Q' = \text{id}_{\text{left}}$
- We define Δ as follows

$$\Delta\Big(f_{w_{\text{left}}}, \varepsilon\Big) = f_{w_{\text{right}}}$$
$$\Delta\Big(f_{w_{\text{left}}}, x\Big) = f_{(wx)_{\text{left}}}$$
$$\Delta\Big(f_{w_{\text{right}}}, x\Big) = f_{(xw)_{\text{right}}}$$

- $F' = \{f_w \mid f_w(q_0) \in F\}$

This works as our automata non-deterministically chooses the number of cyclic shifts and checks if the word formed by undoing them is accepted by the language as if $f_w(q_0) \in F \Rightarrow \hat{\delta}(q_0, w) \in F \Rightarrow w \in L$.

Thus, cyclic(R) is regular $\Rightarrow$ context-free.

(b) Let $R = (V, T, P, S)$ be the generating rules for G in Chomsky Normal Form. To construct R' such that $L(R') = \text{cycle}(G)$

Consider a derivation tree of a string xy in grammar R. Follow the path from S to the leftmost symbol of y.

We wish to generate the path in reverse order (bottom to top) and output symbols on opposite sides of the path from which they originally appeared. To do this construct $G' = (V', T, P', S')$, where

- $V' = V \cup \{U' \mid U \in V\} \cup x$
- $S' = x$
- $P' = P, \{C' \to A'B, B' \to CA' \mid A \to BC \in P\}, x \to \varepsilon, \{x \to aA' \mid A \to a \in P\}, S' \to S$

Thus, cyclic(G) is context-free given G is context-free.

Note, cyclic(G) is not gonna be regular given G is context-free as $\{a^n b^n \mid n \geq 0\}$ is context-free but not regular and cyclic of this language intersection b^*a^* (which is regular) is $\{b^n a^n \mid n \geq 0\}$ which is not regular. cyclic(G) is not gonna be regular given G is context-free.

(c)

Proof. FTSOC, let permutation(R) always be regular, given R is regular.

As $(ab)^*$ is regular and so is a^*b^*, thus,

$$\text{permutation}((ab)^*) \cap (a^*b^*) = \{a^n b^n \mid n \geq 0\}$$

is regular, but that is false.

Hence, we have a contradiction. □

Similarly,

Proof. FTSOC, let permutation(R) always be context free, given R is regular.

As $(abc)^*$ is regular and so is $a^*b^*c^*$, thus,

$$\text{permutation}((abc)^*) \cap (a^*b^*x^*) = \{a^n b^n c^n \mid n \geq 0\}$$

is context-free, but that is false.

Hence, we have a contradiction. □

Thus, Given a regular language R, is permutation(R) is neither guaranteed to be regular nor context-free.

(d) As if some language is regular, it is context-free; then by contra positive, it not being context-free implies that it is irregular.

Proof. FTSOC, let $\text{permutation}(\text{G})$ always be context-free, given G is context-free.

As $(ab)^*(cd)^*$ is context-free (it is regular) and $a^*c^*b^*d^*$ is context-free (it is regular) then

$$\text{permutation}((ab)^*(cd)^*) \cap (a^*c^*b^*d^*) = \{a^n c^m b^n d^m \mid n, m \geq 0\}$$

is context free which is false.

Hence, we have a contradiction. □

Thus, Given a regular language G, is $\text{permutation}(G)$ is neither guaranteed to be regular nor context-free.