CS/ECE 374 Spring 2025 Heng An Cheng (hacheng2@illinois.edu)
Homework 4 Problem 1

Solution:

(a) (i) Let Fooling Set:
$$F = \{0^n 1^n \mid n \geq 1\}$$
Let x and y be arbitrary strings in F .
Then $x = 0^i 1^i, y = 0^j 1^j$ for some non-negative integers $i \neq j$
Let $z = 2^{2i}$
Then $xz = 0^i 1^i 2^{2i} \in L, yz = 0^j 1^j 2^{2i} \notin L,$ because $2i \neq 2j$
Thus, F is a fooling set for L.
Because F is infinite, L cannot be regular

(ii) Let Fooling Set:
$$F = \{(02)^n \mid n \geq 0\}$$
Let x and y be arbitrary strings in F .
Then $x = (02)^i, y = (02)^j$ for some non-negative integers $i \neq j$
Let $z = (12)^i$
Then $xz = (02)^i (12)^i \in L, (02)^j (12)^i \notin L,$ because $i \neq j$
Thus, F is a fooling set for L.
Because F is infinite, L cannot be regular

(iii) Let Fooling Set:
$$F = \{0^{\lceil n\sqrt{n} \rceil} \mid n \geq 1\}$$
Let x and y be arbitrary strings in F .
Without loss of generality, $x = 0^i, y = 0^j$ for some integers $i > j \geq 1$
Let $z = 0^{\lceil (i+1)\sqrt{i+1} \rceil - \lceil i\sqrt{i} \rceil}$
Then
$$xz = 0^{\lceil (i+1)\sqrt{i+1} \rceil}$$
On the other hand
$$yz = 0^{\lceil (i+1)\sqrt{i+1} \rceil - \lceil i\sqrt{i} \rceil + \lceil j\sqrt{j} \rceil} \notin L$$
Because: Define $\lceil n\sqrt{n} \rceil = \lceil n^{\frac{3}{2}} \rceil = a_n$ we know that
$$n^{\frac{3}{2}} \leq a_n < n^{\frac{3}{2}} + 1$$

$$a_{n+1} - a_n \geq (n+1)^{\frac{3}{2}} - n^{\frac{3}{2}} + 1 > n^{\frac{3}{2}} + 1 - (n-1)^{\frac{3}{2}} \geq a_n - a_{n-1}$$
$$\Rightarrow a_{n+1} - a_n > a_n - a_{n-1}$$
the difference between consecutive element in S is increasing when n increases.
That is:
$$\lceil (i+1)\sqrt{i+1} \rceil - \lceil i\sqrt{i} \rceil > \lceil i\sqrt{i} \rceil - \lceil j\sqrt{j} \rceil, \quad for \quad i > j \geq 1$$

$$\Rightarrow \lceil i\sqrt{i}\rceil < \lceil (i+1)\sqrt{i+1}\rceil - \lceil i\sqrt{i}\rceil + \lceil j\sqrt{j}\rceil < \lceil (i+1)\sqrt{i+1}\rceil$$

Thus, F is a fooling set for L.

Because F is infinite, L cannot be regular

(b) Define the fooling set for L_k as:

$$F_k = \{0^*(0+1)^k\}$$

For any $x, y \in F_k$, let s be their longest common suffix, so we write:

$$x = 0^i 0s, \quad y = 0^j 1s, \quad i, j \geq 0.$$

Let d be the difference in the number of $0s$ and $1s$ in x and y before s. Choose:

$$z = 0^{\frac{k-n+d}{2}-1} 1^{\frac{k-n-d}{2}} (01)^k.$$

Then, $xz \in L_k$ since its last $2k$ characters have an equal number of $0s$ and $1s$, while $yz \notin L_k$ since it has more $1s$ than $0s$.

Because F_k is infinite, L_k is not regular.

(c) Suppose L is not regular and L' is a finite language, which also implies L' is regular.

Let $U = L \cup L'$, and assume U is regular.

We can derive that:

$$L = U - L'$$

since L' is finite and regular.

From the closure properties of regular languages, since regular languages are closed under difference, L should be regular.

This contradicts the assumption that L is not regular.

Hence, our assumption that $U = L \cup L'$ is regular must be false.

Thus, $U = L \cup L'$ **is not regular.**

■

CS/ECE 374 Spring 2025 Heng An Cheng (hacheng2@illinois.edu)
Homework 4 Problem 2

Solution:

(a) $L(G)$ is any string over $\{a, b\}$ except those of the form

$$a^n b^n \quad \text{(where } n \text{ is a non-negative integer).}$$

$\overline{L(G)}$ is the set of strings of the form

$$a^n b^n \quad \text{(where } n \text{ is a non-negative integer).}$$

Thus, a CFG for $\overline{L(G)}$ is:

$$S \to aSb \mid \epsilon.$$

$S \to aSb$ wraps the string with an a at the front and a b at the back.

$S \to \epsilon$terminates with empty string.

(b) The grammar will be:

$$S \to aSd \mid Y$$

$$Y \to aYc \mid bYd \mid Z$$

$$Z \to bZc \mid \epsilon$$

Because we need to make sure $a + b = c + d$, so we can only generate one pair from $(a, c), (a, d), (b, c), (b, d)$ each time. We also need to make sure the order a before b and d after c.

- S can create the outer pair (a, d) first.

- Y can generate middle pair $(a, c), (b, d)$, since it will still hold the order a before b and d after c.

- Z is the final pair we can generate the inner pair (b, c) or terminates the string by adding ϵ.

(c) The grammar will be:

$$S \to AB \mid 0A0B0$$

$$A \to 00A0 \mid 1$$

$$B \to 0B00 \mid 1$$

- if i, k are even, we can directly divide original string into A, B, otherwise we make $i = i - 1, j = j - 1, k = k - 1$ by generating $0A0B0$ and then divide string into 2 parts A, B.

- A is the left part of the string, so every time we generate left side 00 and the right side 0 to make sure$i = 2 * j_{left}$.

- B is the right part of the string, so every time we generate right side 00 and the left side 0 to make sure$k = 2 * j_{right}$.

- Thus, we maintain the conditon $i + k = 2 * (j_{left} + j_{right})$

■