Assignment 5 (expanded version)

Mohamed Aref — ID 120210132

Question 1

(1) Context-free grammar language

Consider the grammar

$$L(G) = \{\, 0^n 1^m 0^m 1^n \mid m, n \geq 1 \}.$$

We want to convince ourselves that *every* string of the shape $0^n 1^m 0^m 1^n$ (with both exponents at least 1) is generated by G, and that G generates nothing else.

Derivation idea. Intuitively, the grammar first "peels off" one pair of matching outer symbols 0_front and 1_back at a time (the first arrow below), then creates a matching block of the form $0A1$ (second arrow), and finally grows two equal-length chunks 1^m and 0^m inside that sandwich (third arrow).

Formally, for any chosen $m, n \geq 1$,

$$\begin{aligned} S \;&\Rightarrow^* \; 0^{n-1} S\, 1^{n-1} && \text{(remove } n-1 \text{ matching outer pairs)} \\ &\Rightarrow \; 0^{n-1}\, 0A1\, 1^{n-1} && \text{(introduce the pattern } 0A1\text{)} \\ &\Rightarrow \; 0^n 1^{m-1} A\, 0^{m-1} 1^n && \text{(grow the inner symmetric blocks)} \\ &\Rightarrow \; 0^n 1^m 0^m 1^n. && \text{(finish the inner growth)} \end{aligned}$$

Hence every string that looks like $0^n 1^m 0^m 1^n$ with $m, n \geq 1$ *can* be derived, so $\{0^n 1^m 0^m 1^n \mid m, n \geq 1\} \subseteq L(G)$.

The reverse inclusion (i.e. showing that the grammar cannot generate any other shape) is routine: one proves by induction on derivation length that whenever $S \Rightarrow^* w$ and w is terminal, then w must have exactly the four blocks $0^n 1^m 0^m 1^n$ in that order with equal middle lengths. We omit the mechanical details.

(2) Regular language description

The second grammar generates

$$L(G) = \{\, x \in \{0,1\}^+ \mid x \text{ never contains the substring "00"} \}.$$

Equivalently, $L(G)$ is the set of all non-empty binary strings in which every 0 is immediately followed by a 1. A deterministic finite automaton with two states accepts precisely that language.

Question 2

For each sub-problem we list the productions first, then explain *which* family of strings each group produces.

(1) Three related families of strings

$$\begin{array}{llll} S \rightarrow aS_1 & S_1 \rightarrow bS_1c \mid bc & \text{(generates } \boldsymbol{a\,b^n c^n}) \\ S \rightarrow aS_2c & S_2 \rightarrow aS_2c \mid b & \text{(generates } \boldsymbol{a^n b\,c^n}) \\ S \rightarrow S_3c & S_3 \rightarrow aS_3b \mid ab & \text{(generates } \boldsymbol{a^n b^n c}) \end{array}$$

- *First line* — by pumping S_1 we obtain exactly the strings ab^nc^n with $n \geq 1$.
- *Second line* — each extra application of the aS_2c loop adds one a on the left and one c on the right, keeping the single b in the middle fixed.
- *Third line* — symmetrically adds matching a/b pairs while appending the final c only once at the very end.

(2) A binary palindrome-like pattern

$$S \rightarrow 0S1 \quad | \quad 01 \quad | \quad 0A1, \qquad A \rightarrow 1A \mid 1.$$

The core idea is that S grows outwards: every step inserts a new 0 at the front and a new 1 at the back. We stop either with the shortest legal string 01 or by switching to the non-terminal A, which appends a *positive* number of extra 1s in the middle before the final closing 1. Thus the language is

$$\{\,0^n 1^k 1^n \mid n \geq 1,\ k \geq 1\}.$$

Question 3

Below we simply restate each grammar, then informally describe the language it defines.

(1) Strings over $\{a, b\}$ that start with a

$$S \rightarrow aS_1, \qquad S_1 \rightarrow aS_1 \mid bS_1 \mid a \mid b.$$

After an obligatory leading a (the production $S \rightarrow aS_1$), the non-terminal S_1 may repeatedly copy either a or b and eventually terminate with one single

a or b. Hence $L(G) = \{a\,\{a,b\}^+\}$, i.e. *all non-empty strings of as and bs that begin with a.*

(2) Strings of the form $a^+b^+c^+$

$$S \;\to\; aS_1, \qquad S_1 \;\to\; aS_1 \;\mid\; bS_2, \qquad S_2 \;\to\; bS_2 \;\mid\; cS_3, \qquad S_3 \;\to\; cS_3 \;\mid\; c.$$

The grammar enforces the order $\underbrace{a\cdots a}_{\geq 1}\,\underbrace{b\cdots b}_{\geq 1}\,\underbrace{c\cdots c}_{\geq 1}$:

1. S produces the first a and hands control to S_1.
2. S_1 pumps any additional as, then switches to bs.
3. S_2 generates one or more bs, then calls S_3.
4. S_3 produces at least one c and may generate more before stopping.

Therefore $L(G) = \{\, a^+b^+c^+ \,\}$, a classic example of a non-regular but context-free language.