1. Considerate l'alfabeto $\Sigma = \{a, b\}$

   (a) Fornite una grammatica CF per il linguaggio delle stringhe palindrome di lunghezza pari su Σ, cioè dell'insieme $\text{PAL}_{pari} = \{ww^R \mid w \in \Sigma^*\}$

   Solution: Se si intende $|w|$ pari allora

   $$S \to \epsilon \mid aaSaa \mid bbSbb \mid abSba \mid baSab$$

   oppure

   $$\begin{aligned} S &\to \epsilon \mid aAa \mid bAb \\ B &\to aSa \mid bSb \end{aligned}$$

   Altrimenti se si intende $|ww^R|$ pari allora

   $$S \to \epsilon \mid aSa \mid bSb$$

   (b) Modificate la grammatica precedente per generare l'insieme PAL di tutte le stringhe palindrome su Σ

   Solution:

   $$S \to \epsilon \mid a \mid b \mid aSa \mid bSb$$

   (c) Per ogni $k \in \{0, \ldots, 3\}$, rispondete alla domanda "il linguaggio PAL è di tipo k?" giustificando la risposta.

   Solution: $G_{PAL} = \langle \{S\}, \{a, b\}, \{S \to \epsilon \mid aSa \mid bSb\}, S \rangle$ è una grammatica di tipo 2 (ammettendo le ϵ-regole), quindi PAL è sicuramente un linguaggio context free. Inoltre PAL deve per forza esser riconosciuto con una pila, quindi non può essere regolare.

   (d) Se sostituiamo l'alfabeto con $\Sigma = \{a, b, c\}$, le risposte al punto precedente cambiano? E se lo sostituiamo con $\Sigma = \{a\}$.

   Solution: Nel caso di $\Sigma = \{a, b, c\}$ le risposte non cambiano, infatti basta aggiungere all'alfabeto della pila dell'automa per riconoscere il linguaggio precedente il simbolo c.

Nel caso $\Sigma = \{a\}$ una stringa palindroma su questo alfabeto è banalmente un numero pari di a, questo può essere descritto dalla grammatica regolare (ammenttendo ϵ-regole all'assioma)

$$\begin{aligned} S &\to aA \mid \epsilon \\ A &\to aS \end{aligned}$$

Quindi il linguaggio è di tipo 3.

2. Considerate l'alfabeto $\Sigma = \{a, b\}$. Scrivete una grammatica per generare il complemento di PAL.

Solution: Il complemento di PAL sono tutte le stringhe su $w \in \Sigma^*$ non palindrome.

$$\begin{aligned} S &\to aSa \mid bSb \mid A \\ A &\to aBb \mid bBa \\ B &\to S \mid a \mid b \mid \epsilon \end{aligned}$$

Infatti per finire una stringa bisogna per forza passare per A, che introduce inevitabilmente asimmetria.

3. Sia $\Sigma = \{(,)\}$ l'alfabeto i cui simboli sono la parentesi aperta e la parentesi chius.
   (a) Scrivete una grammatica context-free che generi il linguaggio formato da tutte le sequenze di parentesi correttamente bilanciate, come ad esempio $(()(()))()$.

   Solution:

$$S \to \epsilon \tag{1}$$
$$S \to (S) \tag{2}$$
$$S \to SS \tag{3}$$

$$\begin{aligned}
S &\Rightarrow_3 SS \\
&\Rightarrow_2 (S)S \\
&\Rightarrow_3 (SS)S \\
&\Rightarrow_2 ((S)S)S \\
&\Rightarrow_1 (()S)S \\
&\Rightarrow_2 (()(S))S \\
&\Rightarrow_2 (()((S)))S \\
&\Rightarrow_1 (()(()))S \\
&\Rightarrow_2 (()(()))(S) \\
&\Rightarrow_1 (())(())()
\end{aligned}$$

(b) Risolvete il punto precedente per un alfabeto con due tipi di parentesi, come $\Sigma = \{(,),[,]\}$, nel caso non vi siano vincoli tra i tipi di parentesi (le tonde possono essere contenute tra le quadre e viceversa). Esempio [()([])[]], ma non [[][(])()].

Solution:

$$\begin{aligned}
S &\to \epsilon && (1)\\
S &\to (S) && (2)\\
S &\to [S] && (3)\\
S &\to SS && (4)
\end{aligned}$$

$$\begin{aligned}
S &\Rightarrow_3 [S] \\
&\Rightarrow_4 [SS] \\
&\Rightarrow_4 [SSS] \\
&\Rightarrow_2 [(S)SS] \\
&\Rightarrow_1 [()SS] \\
&\Rightarrow_2 [()(S)S] \\
&\Rightarrow_3 [()([S])S] \\
&\Rightarrow_1 [()([])S] \\
&\Rightarrow_3 [()([])[S]] \\
&\Rightarrow_1 [()([])[]]
\end{aligned}$$

(c) Risolvete il punto precedente con $\Sigma = \{(,),[,]\}$, con il vincolo che le parentesi quadre non possano mai apparire all'interno di parentesi tonde. Ad esempio [()(())[][]](()()), ma non [()([])[]].

Solution:

$$\begin{align}
S &\to [S] \tag{1}\\
S &\to T \tag{2}\\
S &\to SS \tag{3}\\
T &\to \epsilon \tag{4}\\
T &\to (T) \tag{5}\\
T &\to TT \tag{6}\\
&\tag{7}
\end{align}$$

Per generare delle parentesi tonde bilanciate bisogna per forza entrare in T, ed una volta entrati non si possono più generare quadre.

$$\begin{aligned}
S &\Rightarrow_3 SS\\
&\Rightarrow_1 [S]S\\
&\Rightarrow_3 [SS]S\\
&\Rightarrow_3 [SSS]S\\
&\Rightarrow_3 [SSSS]S\\
&\Rightarrow_2 [TSSS]S\\
&\Rightarrow_5 [(T)SSS]S\\
&\Rightarrow_4 [()SSS]S\\
&\Rightarrow_2 [()TSS]S\\
&\Rightarrow_4 [()(T)SS]S\\
&\Rightarrow_4 [()((T))SS]S\\
&\Rightarrow_3 [()(())SS]S\\
&\Rightarrow_1 [()(())[S]S]S\\
&\Rightarrow_2 [()(())[T]S]S\\
&\Rightarrow_4 [()(())[]S]S\\
&\Rightarrow_1 [()(())[][S]]S\\
&\Rightarrow_2 [()(())[][T]]S\\
&\Rightarrow_4 [()(())[][]]S\\
&\Rightarrow_2 [()(())[][]]T\\
&\Rightarrow_5 [()(())[][]](T)\\
&\Rightarrow_6 [()(())[][]](TT)\\
&\Rightarrow_5 [()(())[][]]((T)T)\\
&\Rightarrow_4 [()(())[][]](()T)\\
&\Rightarrow_5 [()(())[][]](()(T))\\
&\Rightarrow_4 [()(())[][]](()())
\end{aligned}$$

4. Sia $G = \langle V = \{S, B, C\}, \Sigma = \{a, b, c\}, P, S\rangle$, con produzioni

$$\begin{aligned} S &\to aSBC \\ S &\to aBC \\ CB &\to BC \\ aB &\to ab \\ bB &\to bb \\ bC &\to bc \\ cC &\to cc \end{aligned}$$

Dopo avere stabilito di che tipo è G, provate a derivare alcune stringhe. Riuscite a dire da quali stringhe è formato il linguaggio generato da G?.

Solution: La grammatica è di tipo 1, visto che nelle parti sinistre delle regole ci sono più elementi, ma i lati destri crescono in lunghezza.

Provando a fare alcune derivazioni otteniamo

$$
\begin{aligned}
S &\Rightarrow aBC \\
&\Rightarrow abC \\
&\Rightarrow abc \\
S &\Rightarrow aSBC \\
&\Rightarrow aaBCBC \\
&\Rightarrow aaBBCC \\
&\Rightarrow aabBCC \\
&\Rightarrow aabbCC \\
&\Rightarrow aabbcC \\
&\Rightarrow aabbcc \\
S &\Rightarrow aSBC \\
&\Rightarrow aaSBCBC \\
&\Rightarrow aaaBCBCBC \\
&\Rightarrow aaaBCBBCC \\
&\Rightarrow aaaBBCBCC \\
&\Rightarrow aaaBBBCCC \\
&\Rightarrow aaabBBCCC \\
&\Rightarrow aaabbBCCC \\
&\Rightarrow aaabbbCCC \\
&\Rightarrow aaabbbcCC \\
&\Rightarrow aaabbbccC \\
&\Rightarrow aaabbbccc
\end{aligned}
$$

Quindi genera il linguaggio $L = \{a^n b^n c^n \mid n > 0\}$.

5. Sia $G = \langle V = \{S, B, C\}, \Sigma = \{a, b, c\}, P, S\rangle$, con produzioni

$$S \rightarrow aBSc \tag{1}$$
$$S \rightarrow abc \tag{2}$$
$$Ba \rightarrow aB \tag{3}$$
$$Bb \rightarrow bb \tag{4}$$

Dopo aver stabilità di che tipo è G, provate a derivare alcune stringhe. Riuscite a dire quali da stringhe formato il linguaggio generato da G?

Solution: La grammatica è di tipo 1, visto che nelle parti sinistre delle regole ci sono più elementi, ma i lati destri crescono in lunghezza.

Provando a fare alcune derivazioni otteniamo

$$
\begin{aligned}
S &\Rightarrow_2 abc \\
S &\Rightarrow_1 aBSc \\
&\Rightarrow_2 aBabcc \\
&\Rightarrow_3 aaBbcc \\
&\Rightarrow_4 aabbcc \\
S &\Rightarrow_1 aBSc \\
&\Rightarrow_1 aBaBScc \\
&\Rightarrow_2 aBaBaBabcccc \\
&\Rightarrow_3 aaBBaBabcccc \\
&\Rightarrow_3 aaBaBBabcccc \\
&\Rightarrow_3 aaaBBBabcccc \\
&\Rightarrow_3 aaaBBaBbcccc \\
&\Rightarrow_3 aaaBaBBbcccc \\
&\Rightarrow_3 aaaaBBBbcccc \\
&\Rightarrow_4 aaaaBBbbcccc \\
&\Rightarrow_4 aaaaBbbbcccc \\
&\Rightarrow_4 aaaabbbbcccc
\end{aligned}
$$

Quindi abbiamo ancora il linguaggio $L = \{a^n b^n c^n \mid n > 0\}$.

6. Sia $G = \langle V = \{S, A, B, C, D, E\}, \Sigma = \{a, b\}, P, S\rangle$, con P

$$
\begin{align}
S &\to ABC \tag{1}\\
AB &\to aAD \tag{2}\\
AB &\to bAE \tag{3}\\
DC &\to BaC \tag{4}\\
EC &\to BbC \tag{5}\\
Da &\to aD \tag{6}\\
Db &\to bD \tag{7}\\
Ea &\to aE \tag{8}\\
Eb &\to bE \tag{9}\\
AB &\to \epsilon \tag{10}\\
C &\to \epsilon \tag{11}\\
aB &\to Ba \tag{12}\\
bB &\to Bb \tag{13}
\end{align}
$$

Dopo avere stabilito di che tipo è G, provate a derivare alcune stringhe. Riuscite a dire da quali stringhe è formato il linguaggio generato da G?

Suggerimento. Per ogni $w \in \{a, b\}^*$ è possibile costruire una derivazione $S \overset{*}{\Rightarrow} wABwC$ (provate a procedere per induzione sulla lunghezza di w cercando di capire il ruolo di ciascuna delle variabili nel processo di derivazione).

Solution: Visto che i lati destri in alcuni casi decrescono, le grammatiche sono di tipo 0. (???).

Provando a fare alcune derivazioni otteniamo

$$
\begin{aligned}
S &\Rightarrow_1 ABC\\
&\Rightarrow_{10} C\\
&\Rightarrow_{11} \epsilon\\
S &\Rightarrow_1 ABC\\
&\Rightarrow_2 aADC\\
&\Rightarrow_4 aABaC\\
&\Rightarrow_3 abAEaC\\
&\Rightarrow_8 abAaEC\\
&\Rightarrow_5 abAaBbC\\
&\Rightarrow_{12} abABabC\\
&\Rightarrow_{10} ababC\\
&\Rightarrow_{11} abab
\end{aligned}
$$

Questa genera il linguaggio $L = \{ww \mid w \in \Sigma\}$, infatti per ogni w è possibile costruire $S \overset{*}{\Rightarrow} wABwC$ ed utilizzare le regole 10 e 11 per ottenere $S \overset{*}{\Rightarrow} ww$. Procediamo su induzione sulla lunghezza di w.

- caso base, $w = \epsilon$, basta applicare

$$\begin{aligned} S &\Rightarrow_1 ABC \\ &\Rightarrow_{10} C \\ &\Rightarrow_{11} \epsilon \end{aligned}$$

- passo, supponiamo di poter derivare $wABwC$, allora possiamo aggiungere una a ad entrambe le parole utilizzando la sequenza di derivazioni

$$\begin{aligned} wABwC &\Rightarrow_2 waADwC \\ &\overset{*}{\Rightarrow}_{6,7} waAwDC \\ &\Rightarrow_4 waAwBaC \\ &\overset{*}{\Rightarrow}_{12,13} waABwaC \end{aligned}$$

  e possiamo aggiungere ad entrambe una b utilizzando la sequenza di derivazioni

$$\begin{aligned} wABwC &\Rightarrow_3 wbAEwC \\ &\overset{*}{\Rightarrow}_{8,9} wbAwEC \\ &\Rightarrow_5 wbAwBbC \\ &\overset{*}{\Rightarrow}_{12,13} wbABwbC \end{aligned}$$

  Quindi, chiamando w' la prima parola e w'' la seconda parola, si utilizzando le produzioni 2 o 3 per aggiungere una a o b a w', si attraversa l'interezza di w'' da sinistra a destra fino a C. Se ad attraversare w'' è stato D, allora una volta a C possiamo generare una a, altrimenti se è stato E possiamo generare una b. Ora si "ripassa il testimone" indietro, e si riattraversa w'' con B fino a tornare ad A, in modo tale da poter riapplicare o 2 o 3.

7. Sia $G = \langle V = \{S, A, B, C, X, Y, L, R\}, \Sigma = \{a\}, P, S\rangle$, con P definito come

$$S \to LXR \tag{1}$$
$$LX \to LYYA \tag{2}$$
$$AX \to YYA \tag{3}$$
$$AR \to BR \tag{4}$$
$$YB \to BX \tag{5}$$
$$LB \to L \tag{6}$$
$$LX \to aC \tag{7}$$
$$CX \to aC \tag{8}$$
$$CR \to \epsilon \tag{9}$$

Riuscite a stabilire da quali stringhe è formato il linguaggio generato da G? *Suggerimento.* Si può osservare che $LX^iR \overset{*}{\Rightarrow} LY^{2i}AR \overset{*}{\Rightarrow} LX^{2i}R$ per ogni $i > 0$. Inoltre dal simbolo iniziale si ottiene la forma LXR. Le ultime tre produzioni sono utili per sostituire le variabili un una forma sentenziale con occorrenze di terminali.

Solution: Proviamo prima di tutto a svolgere alcune produzioni

$$\begin{aligned} S &\Rightarrow_1 LXR \\ &\Rightarrow_7 aCR \\ &\Rightarrow_9 a \end{aligned}$$

$$\begin{aligned} S &\Rightarrow_1 LXR \\ &\Rightarrow_2 LYYAR \\ &\Rightarrow_4 LYYBR \\ &\Rightarrow_5 LYBXR \\ &\Rightarrow_5 LBXXR \\ &\Rightarrow_6 LXXR \\ &\Rightarrow_7 aCXR \\ &\Rightarrow_8 aaCR \\ &\Rightarrow_9 aa \end{aligned}$$

Dato LX^iR, abbiamo due casi:

- se i è uno, allora possiano direttamente applicare la regola 2, ottenendo Y^2, per cui la regola vale

- se $i > 1$, allora dopo aver applicato la regola 2, applichiamo una serie di regole 3, che per ogni X genera due nuove Y

Una volta fatto questo si riconvertono tutte le Y in X utilizzando la regola 4 seguita dalla 5. A questo punto si può scegliere se riapplicare la serie di regole predententi o trasformare le X in a. Quindi il linguaggio generato è $L = \{a^{2^i} \mid i > 0\}$.

8. Modificate la grammatica dell'esercizio precedente in modo da ottenere una grammatica di tipo 1 che generi lo stesso linguaggio

Solution:

$$S \to a \tag{1}$$
$$S \to ZB \tag{2}$$
$$B \to \epsilon \tag{3}$$
$$B \to CB \tag{4}$$
$$aC \to Caa \tag{5}$$
$$ZC \to Za \tag{6}$$
$$ZC \to aa \tag{7}$$

O, equivalentemente $Za \to aa$ invece che $ZC \to aa$.

Svolgendo alcune derivazioni si vede che il linguaggio sembra corretto Generiamo $a^{2^0} = a^1 = a$

$$S \Rightarrow_1 a$$

Generiamo $a^{2^1} = a^2 = aa$

$$\begin{aligned} S &\Rightarrow_2 ZB \\ &\Rightarrow_4 ZCB \\ &\Rightarrow_3 ZC \\ &\Rightarrow_7 aa \end{aligned}$$

Generiamo $a^{2^2} = a^4 = aaaa$

$$\begin{aligned} S &\Rightarrow_2 ZB \\ &\Rightarrow_4 ZCB \\ &\Rightarrow_4 ZCCB \\ &\Rightarrow_3 ZCC \\ &\Rightarrow_6 ZaC \\ &\Rightarrow_5 ZCaa \\ &\Rightarrow_7 aaaa \end{aligned}$$

Generiamo $a^{2^3} = a^8 = aaaaaaaa$

$$\begin{aligned} S &\Rightarrow_2 ZB \\ &\Rightarrow_4 ZCB \\ &\Rightarrow_4 ZCCB \\ &\Rightarrow_4 ZCCCB \\ &\Rightarrow_3 ZCCC \\ &\Rightarrow_6 ZaCC \\ &\Rightarrow_5 ZCaaC \\ &\Rightarrow_6 ZaaaC \\ &\Rightarrow_5 ZaaCaa \\ &\Rightarrow_5 ZaCaaaa \\ &\Rightarrow_5 ZCaaaaaa \\ &\Rightarrow_7 aaaaaaaa \end{aligned}$$

Generiamo $a^{2^4} = a^{16} = aaaaaaaaaaaaaaaa$

$$
\begin{aligned}
S &\Rightarrow_2 ZB \\
&\Rightarrow_4 ZCB \\
&\Rightarrow_4 ZCCB \\
&\Rightarrow_4 ZCCCB \\
&\Rightarrow_4 ZCCCCB \\
&\Rightarrow_3 ZCCCC \\
&\Rightarrow_6 ZaCCC \\
&\Rightarrow_5 ZCaaCC \\
&\Rightarrow_6 ZaaaCC \\
&\Rightarrow_5 ZaaCaaC \\
&\Rightarrow_5 ZaCaaaaC \\
&\Rightarrow_5 ZCaaaaaaC \\
&\Rightarrow_6 ZaaaaaaaC \\
&\Rightarrow_5 ZaaaaaaaCaa \\
&\Rightarrow_5 ZaaaaaaCaaaa \\
&\Rightarrow_5 ZaaaaaCaaaaaa \\
&\Rightarrow_5 ZaaaaCaaaaaaaa \\
&\Rightarrow_5 ZaaaCaaaaaaaaaa \\
&\Rightarrow_5 ZaaCaaaaaaaaaaaa \\
&\Rightarrow_5 ZaCaaaaaaaaaaaaaa \\
&\Rightarrow_5 ZCaaaaaaaaaaaaaaaa \\
&\Rightarrow_7 aaaaaaaaaaaaaaaaa
\end{aligned}
$$

Analizzando più attentamente si può vedere che passando da destra a sinistra una singola C trasforma a^{2^i-1} in $a^{2\cdot(2^i-1)} = a^{2^{i+1}-2}$ visto che raddoppia ogni a. Infine arrivata alla fine – utilizzando la regola 6 – possiamo eliminare la C da destra e ottenere $a^{2^{i+1}-2+1} = a^{2^{i+1}-1}$.

Infine quando si vuole finire la derivazione il simbolo finale Z viene trasformato in a, così da terminare il processo. Infatti senza Z una volta a destra non si possono più smaltire le C e non si può ottenere una stringa senza variabili.

9. Dimostrate che la grammatica $G = \langle\{A, B, C\}, \{a, b\}, P, S\rangle$, con l'insieme delle produ-

zioni P seguenti

$$
\begin{aligned}
S &\to AB \mid BA \mid A \mid B \\
A &\to aAa \mid aAb \mid bAa \mid bAb \mid a \\
B &\to aBa \mid aBb \mid baB \mid bBb \mid b
\end{aligned}
$$

genera il linguaggio $L = \{w \in \{a, b\}^* \mid \forall x \in \{a, b\}^* w \neq xx\}$.

Solution: Analizzando caso per caso le produzioni di S abbiamo

- $S \to A \mid B$ non può generare una parola formata da due stringhe uguali perché può generare solo parole di dimensioni dispari.
- $S \to AB \mid BA$. Prendiamo il caso AB, e l'altro sarà simmetrico. Visto che A deve avere al centro una a e B una b, l'unico modo per cui si possa ottenere $w = x \cdot x$ è che le stringhe generate da A e B siano di dimensione diversa. Supponiamo per assurdo di avere due stringhe α generata da A e β da B tali per cui $\alpha \cdot \beta = x \cdot x$. Deve valere che

  $$x \cdot x = (\alpha \cdot \underbrace{\delta) \cdot (\gamma \cdot \zeta}_{\beta})$$

  quindi $x = \alpha \cdot \delta = \gamma \cdot \zeta$ e

  $$
  \begin{aligned}
  \alpha &= \gamma \\
  \delta &= \zeta
  \end{aligned}
  $$

  Ora sappiamo che la lettera centrale di α è a perché è generata da A. Invece la lettera centrale di β sarà in posizione

  $$\frac{|\beta|}{2} = \frac{|\delta| + |\gamma| + |\zeta|}{2} = \frac{2 \cdot |\delta| + |\gamma|}{2} = |\delta| + \frac{|\gamma|}{2}$$

  Quindi la lettera centrale di β è per forza anche la lettera centrale di γ, e la lettera centrale di γ è per forza b perché β è generata da B, di conseguenza

  $$\alpha \neq \gamma$$

  Quindi vale sia che che $\alpha \cdot \delta \neq \gamma \cdot \zeta$ e $\alpha \cdot \delta = x = \gamma \cdot \zeta$, che è assurdo.