Es. 1) Sia $\Lambda = \{\texttt{x, y, z, 0, 1, 2}\}$

a) Definire una grammatica $\mathcal{G} = <\Lambda, V, S, P>$ che genera il linguaggio:

$$\mathcal{L} = \{\ \texttt{0}^m\ \texttt{1}^n\ \texttt{2}^m\ \texttt{x}^{k-2}\ \texttt{y}\ \texttt{z}^k \mid m \geqslant 0,\ n \geqslant 1,\ k \geqslant 2\}.$$

Per esempio, la stringa `011112xxxyzzzzz` appartiene a $\mathcal{L}$.

b) Dire se le stringhe $w1$ = "1yzz", $w2$ = "0122xyzzz" e $w3$ = "00122yzz" appartengono al linguaggio $\mathcal{L}$, giustificando la risposta tramite l'albero di derivazione o una sequenza di derivazioni.

c) Dire se la grammatica definita è ambigua. Se no, argomentare la risposta. Se si, dimostrarlo.

d) Dire se le stringhe su Λ che iniziano con una sequenza, non vuota, di ''0'' seguita da una sequenza, non vuota, di ''1'' seguita da una sequenza, non vuota, di ''2'' e terminano con "yzz" e, inoltre, il numero di ''0'' è uguale al numero di ''1'' che è uguale al numero di ''2'' appartengono ad $\mathcal{L}$, ovvero dire se $\{0^n1^n2^nyzz \mid n \geqslant 1\} \subseteq \mathcal{L}$.

Es.2) Dato il linguaggio $\mathcal{L}$ definito nell'esercizio precedente, definire un sistema di transizione per $\mathcal{L}$ in modo che la semantica di una stringa $s \in \mathcal{L}$ sia **(i)** <`ID corretto`> se il numero degli "`1`" contenuti in s è uguale al numero delle "`z`" contenute in s; **(ii)** <`ID errato`> in caso contrario.

Per esempio, la semantica della stringa "`01112xyzzz`" è <`ID corretto`> e
la semantica della stringa "`00122yzz`" è <`ID errato`>.

Le configurazioni del sistema contengono, tra le altre, $\{s \mid s \in \mathcal{L}\}$.
Svolgere l'esercizio specificando anche le restanti configurazioni.

Es. 3) Dire se i seguenti comandi `COM1` e `COM2` sono equivalenti.

[redacted]

sapendo che lo stato Sigma è costituito da {x->4, y->20, z=tt}

COM1: z= (x+20)-(y/5)>20
COM2: z= !(z && ((y*2)-30)==15)

NB: due comandi sono equivalenti se si verifica una delle seguenti condizioni:
1- entrambi non terminano
2- terminano in uno stesso stato

Es. 1) Sia Λ={Come, Sempre, Questo, È, Un, Compito, Già, Fatto, Ma, Prudenza} e siano
$\mathcal{L}_0 = \{\texttt{Come}^n\ \texttt{Sempre}^m \mid n \geqslant 2,\ m \geqslant 1\}$,
$\mathcal{L}_1 = \{\texttt{Questo}^n\ (\texttt{È Un})^k\ \texttt{Compito}^n \mid k \geqslant 0,\ n \geqslant 1\}$ e
$\mathcal{L}_2 = \{\texttt{Già}^n\ \texttt{Fatto}^{n+m}\ (\texttt{Ma Prudenza})^m \mid n \geqslant 1,\ m \geqslant 2\}$ linguaggi su Λ.

a) Definire una grammatica che genera il linguaggio:

$$\mathcal{L}_{012} = \{(s_0 s_1)^n \mid n \geqslant 0,\ s_0 \in \mathcal{L}_0 \text{ e } s_1 \in \mathcal{L}_1\} \bigcup \{(s_1 s_2)^n \mid n \geqslant 0,\ s_1 \in \mathcal{L}_1 \text{ e } s_2 \in \mathcal{L}_2\}$$

Per esempio, la stringa `ComeComeSempreQuestoÈUnÈUnCompito` appartiene a $\mathcal{L}_{012}$.

b) La stringa `QuestoCompitoGiàFattoFattoFattoMaPrudenzaMaPrudenza` appartiene a $\mathcal{L}_{012}$? Se no, motivare la risposta. Se si, mostrare l'albero di derivazione.

c) Dire se la grammatica definita è ambigua. Se no, argomentare la risposta. Se si, dimostrarlo.

Es.2) Sia A_1={a, ..., z} l'insieme di tutte le lettere dell'alfabeto e sia A_2={0, ..., 9} l'insieme di tutte le cifre numeriche. Sia A_1^+ l'insieme di tutte le stringhe su A_1 ad esclusione della stringa ε e sia A_2^+ l'insieme di tutte le stringhe su A_2 ad esclusione della stringa ε. Definire un sistema di transizione deterministico, in modo che la semantica di una stringa $s \in \{s_1 s_2 \mid s_1 \in A_1^+ \text{ e } s_2 \in A_2^+\}$ sia $<s_2 s_1>$, dove s_2 è la sequenza delle cifre numeriche in s, e s_1 è la sequenza delle lettere dell'alfabeto in s, <u>invertite</u>.

Per esempio, la semantica della stringa "`bcbaaaccaabaacabbb873549502740`" è <`873549502740bbbacaabaaccaaabcb`>.

Le configurazioni del sistema contengono, <u>tra le altre</u>, $\{s \mid s \in A^+\}$.
Svolgere l'esercizio <u>specificando anche le restanti configurazioni</u>.

ESERCIZIO 3:

Dato lo stato σ [34/x] dove σ= φ1.φ2.Ω e φ1={x->23, y->45} e φ2={x->12,z=tt}

Si determini lo stato finale dell'esecuzione in sequenza dei due comandi:

COM1: z= (x>y) || z

COM2: y= y/((x/2)-17)

Suggerimento: determinare il nuovo stato raggiunto dopo COM1 e usarlo per eseguire COM2