Es. 1. 2019

(a.) $\Sigma = \{a, b\}$

		a	b
$\rightarrow$	q_0	q_0	q_1
	q_1	q_2	q_3
	q_2	q_2	q_3
*	q_3	q_2	q_4
	q_4	q_0	q_1

$\{q_0, q_1, q_2, q_4\}$ $\{q_3\}$

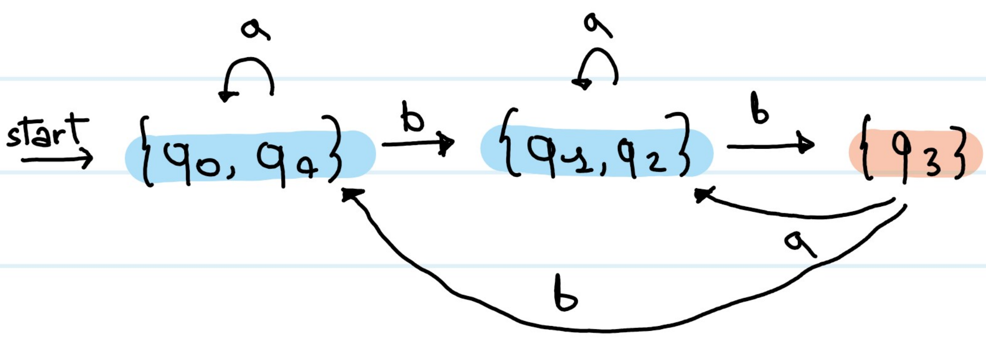
a: $\{q_0, q_1, q_2, q_4\}$ $\{q_3\}$

b: $\{q_0, q_4\}$ $\{q_1, q_2\}$ $\{q_3\}$

$\{q_0, q_4\}$ $\{q_1, q_2\}$ $\{q_3\}$

a: $\{q_0, q_4\}$ $\{q_1, q_2\}$ $\{q_3\}$

b: $\{q_0, q_4\}$ $\{q_1, q_2\}$ $\{q_3\}$



(b.) $P = \{ \{q_0, q_4\} \to a\{q_0, q_4\} \mid b\{q_1, q_2\},$

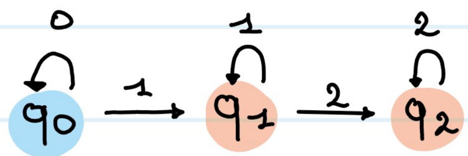
$\{q_1, q_2\} \to a\{q_1, q_2\} \mid b\{q_3\},$

$\{q_3\} \to a\{q_1, q_2\} \mid b\{q_0, q_4\} \mid \varepsilon \}$

$$G = (\underbrace{\{\{q_0, q_4\}, \{q_1, q_2\}, \{q_3\}\}}_{V}, \underbrace{\{a, b\}}_{T}, P, \underbrace{\{q_0, q_4\}}_{E})$$

Es. 2.2019

(i) $L_1 = \{0^n 1^m 2^k \mid n \geq 0, m > 0, k \geq 0\}$



accetta L_1, quindi: L_1 è regolare; in quanto regolare, L_1 è anche libero.

(ii) $L_2 = \{0^n 1^m 2^{n+m} \mid n \geq 1, m \geq 0\}$

Se regolare, per il Pumping lemma $\exists\, w = xyz \mid |xy| \leq \overbrace{m}^{\text{n° stati}}$, $y \neq \varepsilon$, $xy^i z \in L_2\ \forall i \in \mathbb{N}$. y è composizione non vuota di 0; xz deve essere accettato, ma così facendo vi sono meno 0 e la stringa viola la condizione di L_2, ↯. Quindi L_2 non è regolare.

$$P \begin{cases} E \to 0I2 \mid 0E2 \\ I \to \varepsilon \mid 1I2 \end{cases}$$ accetta L_2, quindi L_2 è libero.

(iii) $L_3 = \{0^n 1^{n+1} 2^{n+2} \mid n \geq 0\}$

Se libero, per il Pumping lemma $\exists\, w = abcde \mid |bcd| \leq m$, $bd \neq \varepsilon$, $ab^icd^ie \in L_3\ \forall i \in \mathbb{N}$. bcd non può contenere contemporaneamente 0 e 2:

- se non contiene 0, allora b e d contengono solo 1 e 2; ace deve essere accettata da L_3, ma rimuovendo b e d la condizione di L_3 viene violata perché vengono rimossi 1 e 2, mentre rimangono costanti gli 0, ↯.
- se non contiene 2, allora b e d contengono solo 0 e 1; ace deve essere accettata da L_3, ma rimuovendo b e d la condizione di L_3 viene violata perché vengono rimossi 0 e 1 mentre rimangono costanti i 2, ↯.

Es. 3. 2019

```
int n_greater (int A[], int n, int k) {
    if (m == 0 || A[0] >= k) {
        return n;
    } else if (n == 1) {
        return 0;
    } else {
        return n_greater (A+1, n-1, k);
    }
}
```

Es. 5. 2019

$$prefix(L) = \{ w \in L \mid \exists x \neq \varepsilon \text{ t.c. } wx \in L \}$$

L regolare $\Rightarrow$ prefix(L) regolare. Dato un DFA D che accetta L, si può costruire un DFA D' che ne copi tutto eccetto gli stati finali, che diventano gli stati di D attraverso cui almeno uno stato finale è raggiungibile. D' accetta prefix(L).