

1° parte

Classificare nella gerarchia di Chomsky i seguenti linguaggi motivando formalmente la risposta¹:

- (12pt) Classificare al variare di $m \geq 2$ i seguenti linguaggi sull'alfabeto $\{0, 1\}$:

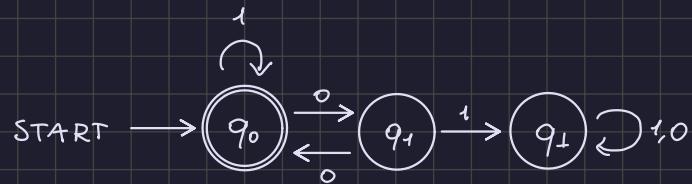
$$A_m = \left\{ \sigma \in \{0, 1\}^* \mid \begin{array}{l} \text{Se } \sigma \text{ contiene 0 allora} \\ \text{tutte le sequenze } 0^n \text{ in } \sigma \text{ sono tali che } n = m \end{array} \right\},$$

Ad esempio:

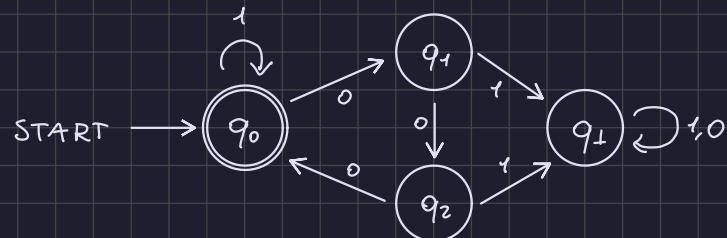
$$\varepsilon, 11111, 110001000, 00011110001, 000 \in A_3, \text{ mentre } 0, 100011110, 1000110000 \notin A_3$$

L'insieme di linguaggi A_m è intuitivamente un insieme di linguaggi regolari, perché per ogni istanza di m non ci sono dipendenze tra gruppi di simboli e quindi i linguaggi sono riconosciuti da una macchina a stati finiti. Alcuni esempi sono i seguenti:

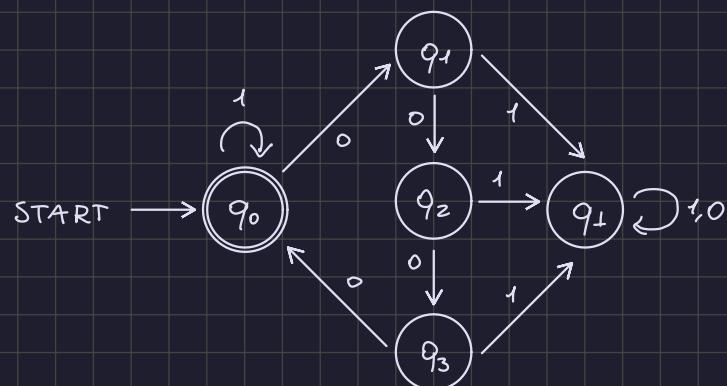
- $m = 2$



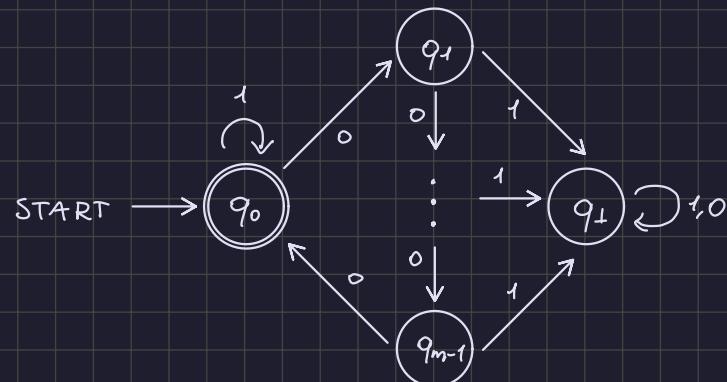
- $m = 3$



- $m = 4$



- $\forall m \geq 2$



Per riconoscere il linguaggio A_m si ha una utoma con $m+1$ stati in cui lo stato iniziale q_0 è l'unico stato finale e riconosce le stringhe 1^* , gli stati dal q_1 al $q_{(m-1)}$ riconoscono la sequenza di zeri lunga m e ogni stato da q_1 a $q_{(m-1)}$ è collegato ad uno stato pozzo.

Procedo a dimostrare l'automa per $m = 2$ e la dimostrazione per $m > 2$ è analoga, ma con uno stato in più.

Dimostrazione

Per dimostrare che un linguaggio è riconosciuto da una macchina a stati deve valere l'uguaglianza:

$$\text{Tesi: } L = L(M) \Leftrightarrow \hat{\delta}(q_0, \sigma) = q_0 \in F$$

Questa tesi si può dividere in 3 casi:

$$\text{Tesi 1: } \sigma \in L \wedge \text{ se contiene 0, tutte le sequenze di 0 sono lunghe 2} \Rightarrow \hat{\delta}(q_0, \sigma) = q_0 \in F$$

$$\text{Tesi 2: } \sigma \notin L \wedge \text{ se contiene 0, tutte le sequenze di 0 sono lunghe 2 e finisce con 10 oppure 000} \Rightarrow \hat{\delta}(q_0, \sigma) = q_1 \notin F$$

$$\text{Tesi 3: } \sigma \notin L \wedge \sigma \text{ contiene una sequenza di 0 non lunga 2} \Rightarrow \hat{\delta}(q_0, \sigma) = q_1 \notin F$$

Dimostro per induzione sulla lunghezza della stringa:

Caso base

Considero le stringhe di lunghezza minima che sono oppure no nel linguaggio:

$$|\sigma|=1 \Rightarrow \begin{cases} \sigma = 1 \Rightarrow \hat{\delta}(q_0, 1) = q_0 \in F \Rightarrow 1 \in L \\ \sigma = 0 \Rightarrow \hat{\delta}(q_0, 0) = q_1 \notin F \Rightarrow 0 \notin L \end{cases}$$

Passo induttivo

Suppongo che le tesi valgano per le stringhe di lunghezza minore o uguale di n e dimostro che valgono per stringhe di lunghezza $n+1$:

Ipotesi induttiva 1: $\exists n \in \mathbb{N}. \forall \sigma \in \{0, 1\}^*. |\sigma| \leq n. ($

$$\sigma \in L \wedge \text{ se contiene 0, tutte le sequenze di 0 sono lunghe 2} \Rightarrow \hat{\delta}(q_0, \sigma) = q_0 \in F$$

Ipotesi induttiva 2: $\exists n \in \mathbb{N}. \forall \sigma \in \{0, 1\}^*. |\sigma| \leq n. ($

$$\sigma \notin L \wedge \text{ se contiene 0, tutte le sequenze di 0 sono lunghe 2 e finisce con 10 oppure 000} \Rightarrow \hat{\delta}(q_0, \sigma) = q_1 \notin F$$

)

Ipotesi induttiva 3: $\exists n \in \mathbb{N}. \forall \sigma \in \{0, 1\}^*. |\sigma| \leq n. ($

$$\sigma \notin L \wedge \sigma \text{ contiene una sequenza di 0 non lunga 2} \Rightarrow \hat{\delta}(q_0, \sigma) = q_1 \notin F$$

Sia σ una stringa nel di lunghezza n , allora esiste una stringa σ' di lunghezza $n+1$ composta da σ e un carattere in più v :

$$\exists n \in \mathbb{N}. \exists \sigma \in L. |\sigma| = n \Rightarrow \exists v \in \{0, 1\}. \sigma' = \sigma v \Rightarrow |\sigma'| = n+1$$

Dimostro che per la stringa σ' valgono le tesi:

- Se $\sigma' \in L$ si distinguono i seguenti casi:

- Se a σ aggiungo 0, per far rimanere σ' nel linguaggio vuol dire che σ se contiene 0, ha tutte le sequenze di zeri lunghe 2 e finisce con 10 oppure 000

$$\begin{aligned}
 \sigma' = \sigma 0 \wedge \sigma & \text{ se contiene } 0, \text{ tutte le sequenze di } 0 \text{ sono lunghe } 2 \Rightarrow \sigma \notin L \Rightarrow \hat{\delta}(q_0, \sigma') \\
 & \Rightarrow \hat{\delta}(q_0, \sigma 0) \\
 & \Rightarrow \delta(\hat{\delta}(q_0, \sigma), 0) \\
 & \stackrel{\text{ii.2}}{\Rightarrow} \delta(q_1, 0) = q_0 \in F \\
 & \Rightarrow \sigma' \in L
 \end{aligned}$$

- Se a σ aggiungo 1, per far rimanere σ' nel linguaggio vuol dire che σ è già nel linguaggio, cioè tutte le sequenze di zeri (se ne ha) sono lunghe 2

$$\begin{aligned}
 \sigma' = \sigma 1 \wedge \sigma & \in L \Rightarrow \hat{\delta}(q_0, \sigma') \\
 & \Rightarrow \hat{\delta}(q_0, \sigma 1) \\
 & \Rightarrow \delta(\hat{\delta}(q_0, \sigma), 1) \\
 & \stackrel{\text{ii.1}}{\Rightarrow} \delta(q_0, 1) = q_0 \in F \\
 & \Rightarrow \sigma' \in L
 \end{aligned}$$

- Se $\sigma' \notin L$ si distinguono i seguenti casi:

- Se a σ aggiungo 0, per far uscire σ' dal linguaggio sono possibili 2 casi:

1) Se σ non è nel linguaggio, cioè contiene una sequenza di zeri non lunga 2

$$\begin{aligned}
 \sigma' = \sigma 0 \wedge \sigma & \text{ contiene una sequenza di zeri non lunga 2} \Rightarrow \hat{\delta}(q_0, \sigma') \\
 & \Rightarrow \hat{\delta}(q_0, \sigma 0) \\
 & \Rightarrow \delta(\hat{\delta}(q_0, \sigma), 0) \\
 & \stackrel{\text{ii.3}}{\Rightarrow} \delta(q_1, 0) = q_1 \notin F \\
 & \Rightarrow \sigma' \notin L
 \end{aligned}$$

2) Se σ è nel linguaggio, se contiene 0, tutte le sequenze di zeri sono lunghe 2

$$\begin{aligned}
 \sigma' = \sigma 0 \wedge \sigma & \text{ ha tutte le sequenze di zeri sono lunghe 2} \Rightarrow \hat{\delta}(q_0, \sigma') \\
 & \Rightarrow \hat{\delta}(q_0, \sigma 0) \\
 & \Rightarrow \delta(\hat{\delta}(q_0, \sigma), 0) \\
 & \stackrel{\text{ii.1}}{\Rightarrow} \delta(q_0, 0) = q_1 \notin F \\
 & \Rightarrow \sigma' \notin L
 \end{aligned}$$

- Se a σ aggiungo 1, per far uscire σ' dal linguaggio sono possibili 2 casi:

1) Se σ contiene 0, ha tutte le sequenze di zeri lunghe 2 e finisce con 10 oppure 000

$$\begin{aligned}
 \sigma' = \sigma 1 \wedge \sigma & \text{ se contiene } 0, \text{ tutte le sequenze di } 0 \text{ sono lunghe } 2 \Rightarrow \hat{\delta}(q_0, \sigma') \\
 & \text{ e finisce con } 10 \text{ oppure } 000 \\
 & \Rightarrow \hat{\delta}(q_0, \sigma 1)
 \end{aligned}$$

$$\Rightarrow \delta(\hat{\delta}(q_0, \sigma), 1)$$

$$\stackrel{i_2}{\Rightarrow} \delta(q_1, 1) = q_1 \notin F$$

$$\Rightarrow \sigma' \notin L$$

1) Se σ non è nel linguaggio, cioè contiene una sequenza di zeri non lunga 2

$$\sigma' = \sigma_0 \wedge \sigma \text{ contiene una sequenza di zeri non lunga 2} \Rightarrow \hat{\delta}(q_0, \sigma')$$

$$\Rightarrow \hat{\delta}(q_0, \sigma_1)$$

$$\Rightarrow \delta(\hat{\delta}(q_0, \sigma), 1)$$

$$\stackrel{i_3}{\Rightarrow} \delta(q_1, 1) = q_1 \notin F$$

$$\Rightarrow \sigma' \notin L$$

Ho dimostrato quindi che il linguaggio A_2 è regolare, la dimostrazione per A_m con m > 2 è analoga, ma con uno stato in più.

- (6pt) Classificare al variare di $m \geq 3$ i seguenti linguaggi sull'alfabeto $\{0, 1\}$ SENZA dimostrarlo formalmente (Fornire solo automa o grammatica e/o stringa per il Pumping Lemma **commentando brevemente**):

$$B_m = \left\{ \sigma \in \{0, 1\}^* \mid \exists i, j \in \mathbb{N} \text{ tali che } \sigma = 1^i 0^{m^2} 1^j \text{ con } i + m \leq j \right\},$$

Esempio: $00011111, 1100011111 \in B_3$, e $\varepsilon, 11, 000, 11000, 1000111 \notin B$

Il resto sembra non considerare m^2 ma solo m

$$B_3 = \left\{ \sigma \in \{0, 1\}^* \mid \exists i, j \in \mathbb{N}. \sigma = 1^i 0^9 1^j. j \geq i+3 \right\}$$

Questo linguaggio è context free e la grammatica che lo genera è la seguente

$$G = \begin{cases} S \rightarrow 1S1 | A111 \\ A \rightarrow A1 | 0^9 \end{cases}$$

$$B_4 = \left\{ \sigma \in \{0, 1\}^* \mid \exists i, j \in \mathbb{N}. \sigma = 1^i 0^{16} 1^j. j \geq i+4 \right\}$$

Questo linguaggio è context free e la grammatica che lo genera è la seguente

$$G = \begin{cases} S \rightarrow 1S1 | A1^4 \\ A \rightarrow A1 | 0^{16} \end{cases}$$

$$B_m = \left\{ \sigma \in \{0, 1\}^* \mid \exists i, j \in \mathbb{N}. \sigma = 1^i 0^{m^2} 1^j. j \geq i+m \right\}$$

$$G = \begin{cases} S \rightarrow 1S1 | A1^m \\ A \rightarrow A1 | 0^{m^2} \end{cases}$$

Quindi B_m è context free

- (12pt) Classificare il seguente linguaggio sull'alfabeto $\{0, 1\}$:

$$C = \bigcup_{m \in \mathbb{N}} B_m \quad D = \bigcap_{m \in \mathbb{N}} B_m$$

$$D = \{\sigma \in \{0, 1\}^* \mid \forall m \in \mathbb{N}. \exists i, j \in \mathbb{N}. \sigma = 1^i 0^m 1^j. j \geq i + m\} = \emptyset$$

Questo linguaggio è regolare per definizione perché è un insieme finito, in questo caso è vuoto.

$$C = \{\sigma \in \{0, 1\}^* \mid \exists i, j, m \in \mathbb{N}. \sigma = 1^i 0^m 1^j. j \geq i + m\}$$

Questo linguaggio non è context free perché ci sono dipendenze tra più gruppi di simboli e oltre a ciò il numero di zeri dipende da una funzione non lineare che non è gestita dai linguaggi context free. Per dimostrarlo bisogna mostrare che non vale il pumping lemma per i linguaggi context free:

$$\neg PL \Rightarrow C \notin CF$$

Il pumping lemma negato è il seguente:

$$\forall k \in \mathbb{N}. \exists z \in L. |z| \geq k. \exists u, v, w, x, y \in \{0, 1\}^*. \begin{cases} z = uvwxy \\ |vwx| \leq k \wedge \exists i \in \mathbb{N}. uv^iwx^i y \notin L \\ |vx| > 0 \end{cases}$$

Condizioni di appartenenza

Le condizioni di appartenenza al linguaggio sono:

$$\sigma \in L \Leftrightarrow \exists i, j, m \in \mathbb{N}. \sigma = 1^i 0^m 1^j. j \geq i + m$$

Dimostrazione

Sia $k \in \mathbb{N}$ considero la seguente stringa nel linguaggio:

$$z = 1^k 0^{k^2} 1^k \text{ con } i = m = k \text{ e } j = 2k \Rightarrow z = 1^k 0^{k^2} 0^{2k} \Rightarrow |z| = 4k > k$$

Questa stringa è nel linguaggio perché $2k \geq k+k \rightarrow 2k \geq 2k$ ✓

Le suddivisioni possibili per questa stringa sono le seguenti:

	1^k	0^{k^2}	1^{2k}
1.	$v x$		
2.	v	x	
3.	v		x
4.	v		x
5.	v		x
6.	v	x	
7.		$v x$	
8.		v	x
9.	v		x

10.

11.

V X

VX

Le suddivisioni 5 e 6 violano banalmente il pumping lemma perchè le partizioni v , w e x sono lunghe in totale almeno k , quindi viene violato il vincolo per cui $|vwx| \leq k$.

Le suddivisioni 2, 6, 8, 10 in cui ci sono delle partizioni a cavallo tra due gruppi di simboli non sono banalmente nel linguaggio dopo il pompaggio perchè violano il vincolo di ordine.

Per ogni suddivisione cerco un indice di pompaggio che faccia uscire la stringa dal linguaggio:

$$1. \forall x \in \{1^k\}$$

$$\exists i=1^{k+|vx|(i-1)} 0^{k^2} 1^{2k}$$

$$- i=2 \Rightarrow 1^{k+|vx|} 0^{k^2} 1^{2k}$$

Verifico le condizioni di appartenenza

$$z_2 \in L \Leftrightarrow 2k \geq k + |vx|$$

$$\Rightarrow |vx|=0 \text{ Assurdo, viola la condizione del pumping lemma } |vx|>0$$

Quindi $z_2 \notin L$

$$3. \forall v \in 1^k, x \in 0^{k^2}$$

$$\exists i=1^{k+|v|(i-1)} 0^{k^2+|x|(i-1)} 1^{2k}$$

$$- i=2 \Rightarrow 1^{k+|v|} 0^{k^2+|x|} 1^{2k}$$

Verifico le condizioni di appartenenza

$$z_2 \in L \Leftrightarrow 2k \geq k + |v| + |x|$$

$$\Rightarrow |vx|=0 \text{ Assurdo, viola la condizione del pumping lemma } |vx|>0$$

Quindi $z_2 \notin L$

$$7. \forall x \in 0^{k^2}$$

$$\exists i=1^k 0^{k+|vx|(i-1)} 1^{2k}$$

$$- i=2 \Rightarrow 0^k 1^{k+|vx|} 0^{2k}$$

Verifico le condizioni di appartenenza

$$z_2 \in L \Leftrightarrow 2k \geq k + |vx|$$

$$\Rightarrow |vx|=0 \text{ Assurdo, viola la condizione del pumping lemma } |vx|>0$$

Quindi $z_2 \notin L$

2° Parte

Classificare nella teoria matematica della ricorsione i seguenti insiemi ed i loro complementi al variare di $n > 0$ dove necessario:¹ :

1. (6pt-7pt) Studiare gli insiemi $R_n = \text{Range}(\psi_n)$ (e i loro complementari), al variare di $n > 3$, dove:

$$\psi_n(x) = \begin{cases} x^n & 3^x \in W_x \text{ deciso in meno di } n \text{ passi} \\ \uparrow & \text{Altrimenti} \end{cases}$$

$$R_n = \{ x^n \mid$$