Esame 30/01/2024

1° parte

Classificare nella gerarchia di Chomsky i seguenti linguaggi motivando formalmente la risposta[1] :

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

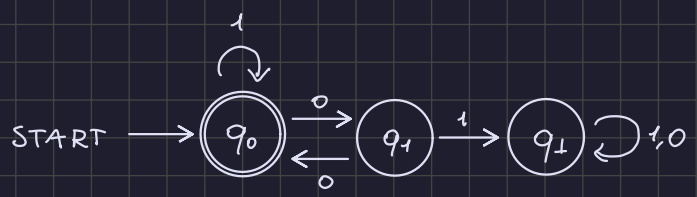
$$A_m = \left\{ \sigma \in \{0,1\}^* \;\middle|\; \begin{array}{l} \text{Se } \sigma \text{ contiene 0 allora} \\ \text{tutte le sequenze } 0^n \text{ in } \sigma \text{ sono tali che } n = m \end{array} \right\},$$

Ad esempio:

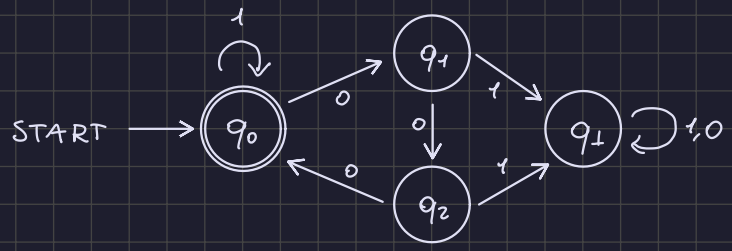
$\varepsilon, 11111, 110001000, 00011110001, 000 \in A_3,$ mentre $0, 1000111110, 1000110000 \notin A_3$

L'insieme di linguaggi A_m è intuitivamente un insieme di linguaggi regolari, perchè 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:

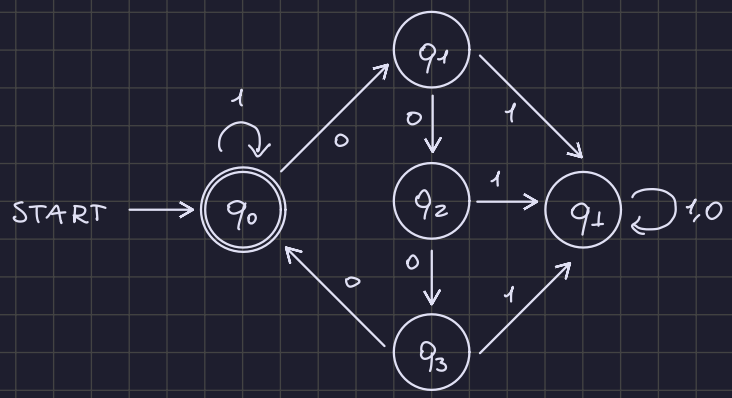
- m = 2



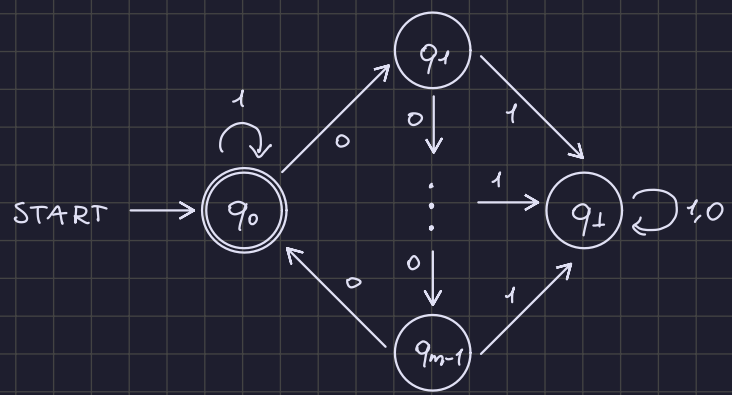
- m = 3



- 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:

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

Questa tesi si può dividere in 3 casi:

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

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

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

Dimostro per induzione sulla lunghezza della stringa:

Caso base

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

$$|x| = 1 \Rightarrow \quad x = 1 \Rightarrow \hat{\delta}(q_0, 1) = q_0 \in F \Rightarrow 1 \in L$$
$$x = 0 \Rightarrow \hat{\delta}(q_0, 0) = q_1 \in F \Rightarrow 0 \notin L$$

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:

$$\text{Ipotesi induttiva 1: } \exists n \in \mathbb{N}. \forall \sigma \in \{0,1\}^*. |\sigma| \leq n. \Big(\sigma \in L \wedge \text{ se contiene 0, tutte le sequenze di 0 sono lunghe 2} \Rightarrow \hat{\delta}(q_0, \sigma) = q_0 \in F \Big)$$

$$\text{Ipotesi induttiva 2: } \exists n \in \mathbb{N}. \forall \sigma \in \{0,1\}^*. |\sigma| \leq n. \Big(\sigma \notin L \wedge \text{ se contiene 0, tutte le sequenze di 0 sono lunghe 2 e finisce con 10 oppure 000} \Rightarrow \hat{\delta}(q_0, \sigma) = q_1 \notin F \Big)$$

$$\text{Ipotesi induttiva 3: } \exists n \in \mathbb{N}. \forall \sigma \in \{0,1\}^*. |\sigma| \leq n. \Big(\sigma \notin L \wedge \sigma \text{ contiene una sequenza di 0 non lunga 2} \Rightarrow \hat{\delta}(q_0, \sigma) = q_\perp \notin F \Big)$$

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

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

$\Rightarrow \hat{\delta}(q_0, \sigma 0)$

$\Rightarrow \delta(\hat{\delta}(q_0, \sigma), 0)$

$\overset{i.2}{\Rightarrow} \delta(q_1, 0) = q_0 \in F$

$\Rightarrow \sigma' \in L$

- 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

$\sigma' = \sigma 1 \wedge \sigma \in L \Rightarrow \hat{\delta}(q_0, \sigma')$

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

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

$\overset{i.1}{\Rightarrow} \delta(q_0, 1) = q_0 \in F$

$\Rightarrow \sigma' \in L$

- 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

    $\sigma' = \sigma 0 \wedge \sigma$ contiene una sequenza di zeri non lunga 2 $\Rightarrow \hat{\delta}(q_0, \sigma')$

    $\Rightarrow \hat{\delta}(q_0, \sigma 0)$

    $\Rightarrow \delta(\hat{\delta}(q_0, \sigma), 0)$

    $\overset{i.3}{\Rightarrow} \delta(q_1, 0) = q_1 \notin F$

    $\Rightarrow \sigma' \notin L$

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

    $\sigma' = \sigma 0 \wedge \sigma$ ha tutte le sequenze di zeri sono lunghe 2 $\Rightarrow \hat{\delta}(q_0, \sigma')$

    $\Rightarrow \hat{\delta}(q_0, \sigma 0)$

    $\Rightarrow \delta(\hat{\delta}(q_0, \sigma), 0)$

    $\overset{i.1}{\Rightarrow} \delta(q_0, 0) = q_1 \notin F$

    $\Rightarrow \sigma' \notin L$

  - 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

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

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

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

$\overset{i.2}{\Rightarrow} \delta(q_1, 1) = q_\perp \notin F$

$\Rightarrow \sigma' \notin L$

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

$\sigma' = \sigma 0 \wedge \sigma$ contiene una sequenza di zeri non lunga 2 $\Rightarrow \hat{\delta}(q_0, \sigma')$

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

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

$\overset{i.3}{\Rightarrow} \delta(q_\perp, 1) = q_\perp \notin F$

$\Rightarrow \sigma' \notin L$

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**):

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

Esempio: $00011111,\ 11000111111 \in B_3$, e $\varepsilon, 11, 000, 11000, 1000111 \notin B$

Il testo sembra non considerare m^2 ma solo m

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

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

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

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

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

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

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

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

Quindi B_m è context free

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

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

$$D = \{\sigma \in \{0,1\}^* \mid \forall m \in \mathbb{N}.\ \exists i, j \in \mathbb{N}.\ \sigma = 1^i 0^{m^2} 1^j.\ j \geq i + m\} = \varnothing$$

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

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

Questo linguaggio non è context free perchè 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:

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

Condizioni di appartenenza

Le condizioni di appartenenza al linguaggio sono:

$$\sigma \in L \iff \exists i, j, m \in \mathbb{N}.\ \sigma = 1^i 0^{m^2} 1^j.\ j \geq i + m$$

Dimostrazione

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

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

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

Le suddivisioni possibili per questa stringa sono le seguenti:

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

10. | | ↓ X

11. | | 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. $VX \in 1^k$

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

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

Verifico le condizioni di appartenenza

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

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

Quindi $z_2 \notin L$

3. $V \in 1^k, X \in 0^{k^2}$

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

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

Verifico le condizioni di appartenenza

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

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

Quindi $z_2 \notin L$

7. $W \in 0^{k^2}$

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

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

Verifico le condizioni di appartenenza

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

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

Quindi $z_2 \notin L$

9. $v \in 0^{k^2}, x \in 0^{2k}$

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

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

Verifico le condizioni di appartenenza

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

$\Rightarrow |x| \leq |v|$

$\Rightarrow |x|-|v| \leq 0 \ \wedge \ |v|+|x| \leq k-1 \equiv |v| \leq k-1-|x|$

$\Rightarrow 0 \geq |x|-|v| \geq |x|-(k-1-|x|)$

$\Rightarrow 0 \geq |x|-k+1+|x|$

$\Rightarrow 2|x|+1-k \leq 0$

$\Rightarrow k \geq 2|x|+1$?

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:

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

$R_n = \{ x^n \mid 3^x \in W_x \text{ deciso in meno di n passi} \}$

Questo insieme è intuitivamente ricorsivo perchè esiste una funzione caratteristica:

$$\Psi_{R_n}(x) = \begin{cases} 1 & 3^x \in W_x \text{ deciso in meno di n passi} \\ 0 & 3^x \notin W_x \text{ deciso in meno di n passi} \end{cases}$$

Per dimostrare che questa funzione è ricorsiva fornisco l'algoritmo:

```
input(x)
z = 3^x

costruisci phi_x
for i = 0 to n-1 {
  esegui il prossimo passo di phi_x(z)
  if phi_x(z) ha terminato {
    return 1
  }
}

return 0
```

Siccome R_n è ricorsivo, anche il suo complemento lo è.

2. **(10pt)** Sia $R = \bigcup_{n>0} \text{Dom}(\psi_n)$, studiare il seguente insieme (ed il suo complementare):

$$S = R \cap \{ y \in \mathbb{N} \mid \varphi_y(3^y) \downarrow \Rightarrow \varphi_y(3^y) \in \{3\}^{\mathbb{N}} \}$$ [2]

$R = \bigcup_{n>0} \{ x \mid 3^x \in W_x \text{ deciso in meno di n passi} \}$

$= \bigcup_{n>0} \{ x \mid P_x(3^x) \text{ termina in meno di n passi} \}$

$= \{ x \mid \exists n>0 . P_x(3^x) \text{ termina in meno di n passi} \}$

$S = \{ x \mid \exists n>0 . P_x(3^x) \text{ termina in meno di n passi} \wedge (\varphi_x(3^x)\downarrow \Rightarrow \varphi_x(3^x) \in \{3\}^{\mathbb{N}}) \}$

$= \{ x \mid P_x(3^x)\downarrow \wedge \varphi_x(3^x) \in \{3\}^{\mathbb{N}} \}$

$\bar{S} = \{ x \mid P_x(3^x)\uparrow \wedge \varphi_x(3^x) \notin \{3\}^{\mathbb{N}} \}$

S è intuitivamente RE e Creativo perchè si può solo decidere quando il programma termina, quindi esiste una funzione semicaratteristica che dice se un elemento è nell'insieme e diverge quando non lo è.

$$f_S(x) = \begin{cases} 1 & x \in S \\ \uparrow & \text{altrimenti} \end{cases}$$

Lo pseudocodice è il seguente:

```
input(x)
z = 3^x

costruisci phi_x
while true {
  esegui il prossimo passo di phi_x(z)
  if phi_x(z) ha terminato e phi_x(z) = 3^N {
    return 1
  }
}
```

Per dimostrare che l'insieme è creativo devo ridurre funzionalmente S a k

$$K \preceq S \equiv \bar{K} \preceq \bar{S}$$

Esiste una funzione parziale ricorsiva:

$$\Psi(x,y) = \begin{cases} 3 & \text{se } x \in K \\ \uparrow & \text{altrimenti} \end{cases}$$

Questa funzione è parziale ricorsiva perchè esiste un algoritmo che ritorna 3 quando il programma converge con se stesso in input e diverge altrimenti. Quindi si può applicare il teorema smn.

$$\overset{smn}{\Rightarrow} \exists g \text{ totale ricorsiva} . \quad \Psi(x,y) = \varphi_{g(x)}(y)$$

Riduzione funzionale

$$-x \in K \overset{def K}{\Rightarrow} \varphi_x(x)\downarrow$$

$$\overset{def \Psi}{\Rightarrow} \forall y.\ \Psi(x,y)\downarrow$$

$$\overset{smn}{\Rightarrow} \forall y.\ \varphi_{g(x)}(y)\downarrow$$

$$\Rightarrow \exists y = 3^x.\ \varphi_{g(x)}(y) = 3 \in \{3\}^{\mathbb{N}}$$

$$\overset{def S}{\Rightarrow} g(x) \in S$$

$$-x \in K \overset{def K}{\Rightarrow} \varphi_x(x)\uparrow$$

$$\overset{def \Psi}{\Rightarrow} \forall y.\ \Psi(x,y)\uparrow$$

$$\overset{smn}{\Rightarrow} \forall y.\ \varphi_{g(x)}(y)\uparrow$$

$$\Rightarrow \nexists y\ \varphi_{g(x)}(y) \in \{3\}^{\mathbb{N}}$$

$$\overset{def S}{\Rightarrow} g(x) \notin S$$

Quindi ho dimostrato che S è Creativo, di conseguenza il suo complemento è produttivo.

3. **(12pt)** Studiare il seguente insieme (ed il suo complementare)):

$$T = \{\, x \mid (\exists n \in \mathbb{N}.\, y = 3^{3n} \;\Rightarrow\; \varphi_x(y)\downarrow) \;\wedge\; (y \in W_x \;\Rightarrow\; \exists n \in \mathbb{N}.\, y = 3^n) \,\},$$

$T = \{x \mid \exists n \in \mathbb{N}.\ 3^{3n} \subseteq W_x \subseteq 3^n\}$

Questo insieme è intuitivamente produttivo perchè bisognerebbe controllare ogni singolo input

Per dimostrarlo mostro che l'insieme T si riduce al complemento di K:

$\overline{K} \preceq T \equiv K \preceq \overline{T}$

Per farlo fornisco una funzione parziale ricorsiva:

$$\Psi(x,y) = \begin{cases} 1 & \text{se } x \in K \vee \exists n.\ y = 3^{3n} \\ \uparrow & \text{altrimenti} \end{cases}$$

Questa funzione è parziale ricorsiva perchè esiste un algoritmo che ritorna 3 quando il programma converge con se stesso in input e diverge altrimenti. Quindi si può applicare il teorema smn.

$\overset{smn}{\Rightarrow} \exists g$ totale ricorsiva . $\Psi(x,y) = \varphi_{g(x)}(y)$

Riduzione funzionale

$- x \in K \overset{def K}{\Rightarrow} \varphi_x(x)\downarrow$

$\overset{def \Psi}{\Rightarrow} \forall y.\ \Psi(x,y)\downarrow$

$\overset{smn}{\Rightarrow} \forall y.\ \varphi_{g(x)}(y)\downarrow$

$\overset{def W_{g(x)}}{\Rightarrow} W_{g(x)} = \mathbb{N} \not\subseteq 3^n$

$\overset{def T}{\Rightarrow} g(x) \notin T$

$- x \notin K \overset{def \Psi}{\Rightarrow} \Psi(x,y)\downarrow \Leftrightarrow \exists n \in \mathbb{N}.\ y = 3^{3n}$

$\overset{smn}{\Rightarrow} \varphi_{g(x)}(y)\downarrow \Leftrightarrow \exists n \in \mathbb{N}.\ y = 3^{3n}$

$\overset{def W_{g(x)}}{\Rightarrow} \exists n \in \mathbb{N}.\ W_{g(x)} = 3^{3n}$

$\Rightarrow 3^{3n} \subseteq W_{g(x)} \subseteq 3^n \equiv 3^{3n} \subseteq 3^{3n} \subseteq 3^n$

$\overset{def T}{\Rightarrow} g(x) \in T$

Quindi l'insieme è produttivo

$\overline{T} = \{x \mid \exists n \in \mathbb{N}.\ 3^{3n} \not\subseteq W_x \not\subseteq 3^n\}$

Anche in questo caso l'insieme è produttivo perchè bisogna controllare ogni input.

$\overline{K} \preceq \overline{T} \equiv K \preceq T$

$$\Psi(x,y) = \begin{cases} 1 & \text{se } x \in K \wedge \exists n.\ y = 3^{3n} \\ \uparrow & \text{altrimenti} \end{cases}$$

- $x \in K \overset{\text{def } K}{\Rightarrow} \varphi_x(x)\downarrow$

$\overset{\text{def } \Psi}{\Rightarrow} \Psi(x,y)\downarrow \Leftrightarrow \exists n \in \mathbb{N}.\ y = 3^{3^n}$

$\overset{\text{smn}}{\Rightarrow} \varphi_{g(x)}(y)\downarrow \Leftrightarrow \exists n \in \mathbb{N}.\ y = 3^{3^n}$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} W_{g(x)} = 3^{3^n}$

$\Rightarrow 3^{3^n} \subseteq W_{g(x)} \subseteq 3^n \equiv 3^{3^n} \subseteq 3^{3^n} \subseteq 3^n$

$\overset{\text{def } T}{\Rightarrow} g(x) \in T$

- $x \notin K \overset{\text{def } \Psi}{\Rightarrow} \forall y\ \Psi(x,y)\uparrow$

$\overset{\text{smn}}{\Rightarrow} \forall y\ \varphi_{g(x)}(y)\uparrow$

$\overset{\text{def } W_{g(x)}}{\Rightarrow} W_{g(x)} = \emptyset$

$\Rightarrow 3^{3^n} \subseteq W_{g(x)} \subseteq 3^n \Rightarrow 3^{3^n} \nsubseteq \emptyset$

$\overset{\text{def } T}{\Rightarrow} g(x) \notin T$

Quindi il complemento di T è produttivo.

4. **(2pt)** Che relazione esiste tra T e l'insieme $\{\ x \mid \{27\}^{\mathbb{N}} \subseteq W_x \subseteq \{3\}^{\mathbb{N}}\ \}$? Motivare formalmente la risposta.

$\{x \mid \exists n \in \mathbb{N}.\ W_x \subseteq 3^n \wedge 3^{3^n} \subseteq W_x\} = T$

I due insiemi sono uguali