Elementi di Teoria della Computazione

Classe: Resto_2 - Prof.ssa Marcella Anselmo

Tutorato

13/06/2022 ore 12:00-14:00

Quinta Esercitazione

a cura della dott.ssa Manuela Flores

Prima Prova Intercorso 14/04/2022: linguaggi regolari

1. Per ognuno dei seguenti linguaggi, indicare se sono regolari o no, giustificando la risposta.

   (a) $X = \{ww \mid w \in \{a\}^*\}$.

   (b) $Y = \{ww^r \mid w \in \{a, b\}^*\}$, dove w^r indica il reverse di una stringa.

Pumping lemma: dimostrare la non regolarità (esempio)

Dimostriamo che $L = \{a^n b^n \mid n \geq 0\}$ **non è regolare**!

Dimostrazione.

Supponiamo per assurdo che L sia regolare. Allora vale il pumping lemma.
Sia p la lunghezza del pumping.

Consideriamo la stringa $s = a^p b^p$.
Ovviamente $s \in L$ e $|s| = 2p$ (soddisfa le ipotesi $|s| \geq p$).

Consideriamo **TUTTE** le possibili fattorizzazioni di $s = a^p b^p$ in 3 stringhe x, y, z con le proprietà delle condizioni: $|xy| \leq p$ e $|y| \geq 1$.

$$\overbrace{a\ a\ a\ a\ a\ a\ a\ a\ a}^{p}\ \overbrace{b\ b\ b\ b\ b\ b\ b\ b\ b}^{p}$$

Lezione 13 pag. 97

Pumping lemma: dimostrare la non regolarità (esempio)

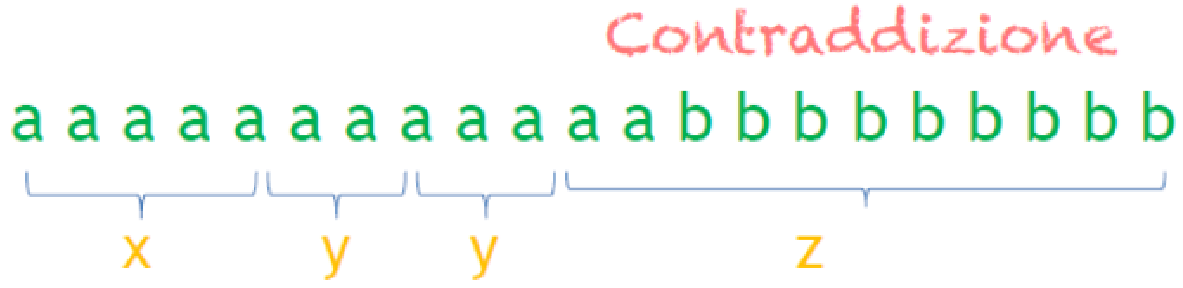
Dimostriamo che $L = \{a^n b^n \mid n \geq 0\}$ **non è regolare**!

Dimostrazione.

...

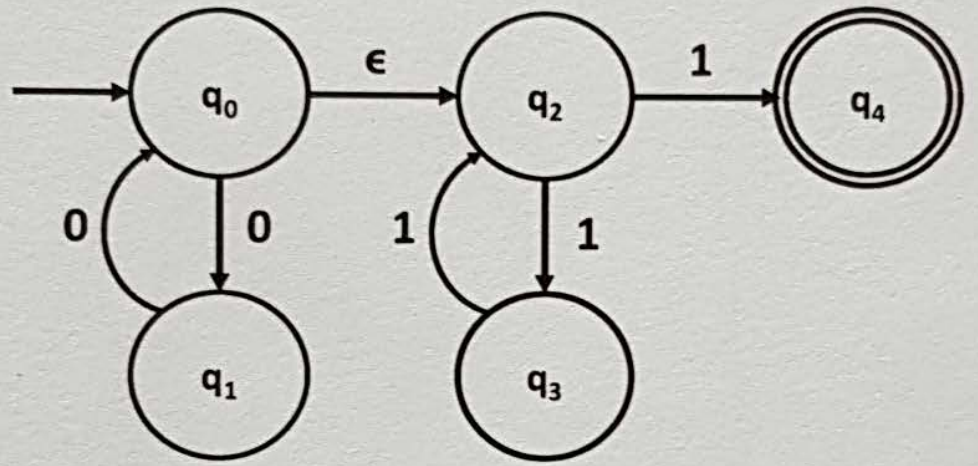
Consideriamo **TUTTE** le possibili fattorizzazioni di $s = a^p b^p$ in 3 stringhe x, y, z con le proprietà delle condizioni: $|xy| \leq p$ e $|y| \geq 1$.

Quindi $y = a^m$, per $1 \leq m \leq p$. Per $i = 2$, $xy^2z = a^{p+m}b^p \notin L$.



Prima Prova Intercorso 14/04/2022: Da NFA a DFA

2. Trasformare il seguente NFA nel DFA equivalente utilizzando la costruzione studiata durante le lezioni. Riportare con precisione la descrizione della funzione di transizione e produrre il diagramma di stato (limitandosi agli stati raggiungibili dallo stato iniziale del DFA). Fornire una espressione regolare che descrive il linguaggio riconosciuto dall'automa.



Prima Prova Intercorso 14/04/2022: espressioni regolari

3. Dimostrare o confutare le seguenti affermazioni. Ricordiamo che se E è una espressione regolare, con $L(E)$ indichiamo il linguaggio descritto da E.

   (a) $L(a^*b^*) \cap L(b^*a^*) = L(a^* \cup b^*)$

   (b) $L((1^*011^*)^*(0 \cup \epsilon) \cup 1^*) =$ insieme delle stringhe binarie che contengono 11.

Espressioni regolari (RE)

DEF[espressione regolare]

Sia Σ un alfabeto,

passo base: le "*espressioni regolari primitive*" sono ϵ, $\emptyset$, ed ogni $a \in \Sigma$.

passo ricorsivo: siano E_1 ed E_2 espressioni regolari, allora

- (E_1) è un'espressione regolare
- $(E_1 \cup E_2)$ è un'espressione regolare
- $(E_1 \cdot E_2)$ (oppure $(E_1 \circ E_2)$ o $(E_1 E_2)$) è un'espressione regolare
- (E_1^*) è un'espressione regolare

Lezione 12 pag. 71

Esercizio (1.53, sipser)

Sia $\Sigma = \{0, 1\}$. Per ogni espressione regolare seguente, indichiamo il linguaggio rappresentato.

1. $0^*10^* = \{w \in \Sigma^* \mid w \text{ contiene un solo } 1\}$
2. $\Sigma^*1\Sigma^* = \{w \in \Sigma^* \mid w \text{ contiene almeno un } 1\}$
3. $\Sigma^*001\Sigma^* = \{w \in \Sigma^* \mid w \text{ contiene la stringa } 001 \text{ come sottostringa}\}$
4. $1^*(01^+)^* = \{w \in \Sigma^* \mid \text{ogni } 0 \text{ in } w \text{ è seguito da almeno un } 1\}$
5. $(\Sigma\Sigma)^* = \{w \in \Sigma^* \mid w \text{ è una stringa di lunghezza pari}\}$
6. $(\Sigma\Sigma\Sigma)^* = \{w \in \Sigma^* \mid \text{la lunghezza di } w \text{ è un multiplo di 3}\}$
7. $01 \cup 10 = \{01, 10\}$
8. $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = \{w \in \Sigma^* \mid w \text{ inizia e termina con lo stesso simbolo}\}$
9. $(0 \cup \epsilon)1^* = 01^* \cup 1^*$
10. $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$
11. $1^*\emptyset = \emptyset$
12. $\emptyset^* = \{\epsilon\}$

Prima Prova Intercorso 14/04/2022: DFA

4. Fornire un DFA che accetta tutte le stringhe su $\{a, b\}$ che non contengono un numero dispari di b e non contengono la stringhe in $\{a\}^+$ come fattore. Fornire una espressione regolare che descrive il linguaggio riconosciuto dall'automa.

Seconda Prova Intercorso 08/06/2022: Computazione di MdT

Quesito 1 (5 punti) (*Computazione di MdT*)

Si consideri la seguente Macchina di Turing, $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, dove $Q = \{ q_0, q_1, q_2, q_3, q_{accept}, q_{reject}, \}$, $\Sigma = \{ a, b\}$, $\Gamma = \{ a, b, \_ \}$ e la funzione δ è definita come segue

$\delta(q_0, a) = (q_1, a, R)$, $\delta(q_0, b) = (q_2, b, R)$, $\delta(q_0, \_) = (q_{reject}, \_ , R)$,

$\delta(q_1, a) = (q_1, a, R)$, $\delta(q_1, b) = (q_1, a, R)$, $\delta(q_1, \_) = (q_{accept}, \_ , R)$,

$\delta(q_2, a) = (q_{reject}, b, R)$, $\delta(q_2, b) = (q_3, b, L)$, $\delta(q_2, \_) = (q_{accept}, \_ , R)$,

$\delta(q_3, a) = (q_{reject}, b, R)$, $\delta(q_3, b) = (q_2, b, R)$, $\delta(q_3, \_) = (q_{reject}, b, R)$.

a) Indicare
- una stringa $\mathbf{w_a}$ di Σ^* che sia **accettata** da M
- una stringa $\mathbf{w_r}$ di Σ^* che sia **rifiutata** da M
- una stringa $\mathbf{w_c}$ di Σ^* su cui M **cicla**

b) Mostrare la **computazione** di M, dalla configurazionale iniziale a una configurazione di arresto, su input $\mathbf{w_a}$ e su input $\mathbf{w_r}$, indicando le configurazioni intermedie e il numero di passi effettuati.

c) **Giustificare** perché M **cicla** su input $\mathbf{w_c}$.

Seconda Prova Intercorso 08/06/2022: MdT che calcola

Quesito 2 (7 punti) (*MdT che calcola*)

Sia $\Sigma = \{ 0, 1 \}$. **Descrivere** una **MdT** deterministica che **calcola** la funzione **f:** $\Sigma^* \rightarrow \Sigma^*$ che ad ogni

w $\in \Sigma^*$ associa la stringa **111** se la lunghezza di w è dispari, **00**, altrimenti.

La descrizione deve essere fornita tramite **settupla** o **diagramma di stato** e deve essere accompagnata da una descrizione **ad alto livello** che ne giustifichi il funzionamento

e non è necessario che si fermi sulla prima cella.

Seconda Prova Intercorso 08/06/2022: Vero o Falso

Quesito 3 (5 punti)

Per ognuna delle seguenti affermazioni dire se è **Vera** o **Falsa**. In entrambi i casi, occorre **motivare** la risposta, citando i risultati noti utilizzati.

a) Se L è **riconosciuto** da una MdT a 2 nastri allora L è riconosciuto da una MdT a singolo nastro.
b) Se L è **riconosciuto in tempo polinomiale** da una MdT a 2 nastri allora L è riconosciuto in tempo polinomiale da una MdT a singolo nastro.
c) A_{TM} è NP-completo.

Lezione 21 pag. 8

Equivalenza fra MdT e MdTM

Il modello di MdT «potenziato» con la possibilità di avere più di un nastro, permette di riconoscere più linguaggi?

Teorema
I due modelli di Mdt e MdTM hanno stesso potere computazionale.

Dimostrazione
In un verso è ovvio: ogni MdT è una MdTM con k=1 nastri.

Viceversa, dimostriamo che per ogni MdT multinastro M esiste una MdT (a singolo nastro) S equivalente ad M, cioè L(S) = L(M).
Ci riferiremo al contenuto (significativo) del nastro.

Lezione 29 pag. 42

La classe P

Teorema

Sia $t(n)$ una funzione tale che $t(n) \geq n$. Per ogni macchina di Turing multinastro M con complessità di tempo $t(n)$ esiste una macchina di Turing a nastro singolo M' con complessità di tempo $O(t^2(n))$, equivalente a M.

Quindi, se L è deciso in tempo polinomiale su una macchina di Turing multinastro, allora L è deciso in tempo polinomiale su una macchina di Turing a nastro singolo.

Lezione 30 pag. 16

La classe NP

Teorema 7.20
Un linguaggio L è in NP se e solo se esiste una macchina di Turing non deterministica che decide L in tempo polinomiale.

Dunque i linguaggi della classe NP sono decidibili:
A_{TM} è indecidibile $\Rightarrow$ NON può essere NP-completo

Definizione
Sia $t : \mathbb{N} \to \mathbb{R}^+$ *una funzione. La classe di complessità in tempo non deterministico* $NTIME(t(n))$ *è*

$$NTIME(t(n)) = \{L \mid \exists \text{ una macchina di Turing non deterministica } M \text{ che decide } L \text{ in tempo } O(t(n))\}$$

Corollario 7.22

$$NP = \bigcup_{k \geq 0} NTIME(n^k)$$

Seconda Prova Intercorso 08/06/2022: Rice

Quesito 4 (6 punti)

a) **Enunciare** il teorema di Rice.
b) Dire se il **teorema di Rice** può essere **applicato** al seguente linguaggio sull'alfabeto $\Sigma = \{ 0, 1\}$, giustificando la risposta. L'eventuale descrizione di MdT può essere data ad alto livello.

$$X = \{ \langle M \rangle \mid M \text{ è una MdT e } M \text{ accetta soltanto stringhe di } \Sigma^* \text{ che finiscono per } 0 \}$$

Lezione 28 pag. 24

Teorema di Rice

Teorema di Rice. Sia

$$L = \{\langle M \rangle \mid M \text{ è una MdT che verifica la proprietà } \mathcal{P}\}$$

un linguaggio che soddisfa le seguenti due condizioni:

1. $\mathcal{P}$ è una proprietà del linguaggio $L(M)$, cioè: prese comunque due MdT M_1, M_2 tali che $L(M_1) = L(M_2)$ risulta

$$\langle M_1 \rangle \in L \Leftrightarrow \langle M_2 \rangle \in L$$

2. $\mathcal{P}$ è una proprietà non banale, cioè: esistono due MdT M_1, M_2 tali che

$$\langle M_1 \rangle \in L, \langle M_2 \rangle \notin L.$$

Allora L è indecidibile.

Seconda Prova Intercorso 08/06/2022: Hampath e Uhampath

Quesito 5 (7 punti)

a) **Definire** i linguaggi HAMPATH e UHAMPATH.
b) Mostrare che **UHAMPATH** appartiene a **NP**.
c) Siano A e B due linguaggi. **Definire** cosa significa che **A $\leq_p$ B**, ovvero che A si riduce in tempo polinomiale a B.

Quesito bonus*

d) Durante il corso abbiamo visto che HAMPATH $\leq_p$ UHAMPATH. Dimostrare adesso che **UHAMPATH $\leq_p$ HAMPATH**.

Lezione 34(b) pag. 2

UHAMPATH

È possibile definire una "versione non orientata" del problema del cammino Hamiltoniano.

- Un cammino Hamiltoniano in un grafo non orientato è un cammino che passa per ogni vertice del grafo una e una sola volta.

$$UHAMPATH = \{\langle G, s, t\rangle \mid G \text{ è un grafo non orientato e ha un cammino Hamiltoniano da } s \text{ a } t\}$$

Per mostrare che *UHAMPATH* è *NP*-completo, definiamo una riduzione di tempo polinomiale da *HAMPATH* a *UHAMPATH*.

Lezione 34(b) pag. 3

UHAMPATH

Teorema

$UHAMPATH \in NP$

Dimostrazione.

Un algoritmo N che verifica *UHAMPATH* in tempo polinomiale:

$N =$ "Sull'input $\langle\langle G, s, t\rangle, c\rangle$, dove $G = (V, E)$ è un grafo non orientato:

1. Verifica se $c = (u_1, \ldots, u_{|V|})$ è una sequenza di $|V|$ vertici di G, altrimenti rifiuta.
2. Verifica se i nodi della sequenza sono distinti, $u_1 = s$, $u_{|V|} = t$ e, per ogni i con $2 \leq i \leq n$, se $(u_{i-1}, u_i) \in E$, accetta in caso affermativo; altrimenti rifiuta."

$\exists c : \langle\langle G, s, t\rangle, c\rangle \in L(N)$ se e solo se $\langle G, s, t\rangle \in UHAMPATH$. $\square$

Lezione 32 pag. 3

Riduzioni in tempo polinomiale

Definizione

Siano A, B linguaggi sull'alfabeto Σ.
Una riduzione in tempo polinomiale f di A in B è

- una funzione $f : \Sigma^* \rightarrow \Sigma^*$
- calcolabile in tempo polinomiale
- tale che per ogni $w \in \Sigma^*$

$$w \in A \Leftrightarrow f(w) \in B$$

Definizione

Un linguaggio $A \subseteq \Sigma^*$ è riducibile in tempo polinomiale a un linguaggio $B \subseteq \Sigma^*$, e scriveremo $A \leq_p B$, se esiste una riduzione di tempo polinomiale di A in B.

Lezione 34(b) pag. 7

$HAMPATH \leq_p UHAMPATH$

Costruzione di G':

- Ogni vertice u di G, diverso da s e t è rimpiazzato da tre vertici u^{in}, u^{mid} e u^{out} in G'.
- I vertici s e t sono sostituiti con i vertici s^{out} e t^{in} in G'.
- Per ogni $u \in V \setminus \{s, t\}$, (u^{in}, u^{mid}) e (u^{mid}, u^{out}) sono in E'.
- Se $(u, v) \in E$ allora $(u^{out}, v^{in}) \in E'$.

Prossimo tutorato

Prima del primo appello di luglio:
data da definire, la troverete pianificata su questo canale del Team…
…buono studio
e in bocca al lupo
per il preappello
di mercoledì prossimo ☺