Elementi di
Teoria della Computazione

Classe: Resto_2 - Prof.ssa Marcella Anselmo

Tutorato
03/07/2022 ore 15:00-17:00

Settima Esercitazione

a cura della dott.ssa Manuela Flores

Linguaggi regolari: dimostrare o confutare

Dimostrare o confutare le seguenti affermazioni.

(a) Il linguaggio $X = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ è regolare (indichiamo con $|w|_a$ e $|w|_b$ rispettivamente le occorrenze di a e di b in w).

(b) Il linguaggio $Y = \{a^i b^j \mid i + j \text{ è multiplo di } 2\}$ è regolare.

(c) $L((ab)^*) \cap L((cd)^*) = \emptyset$, dove $L(E)$ è il linguaggio denotato dall'espressione regolare E.

Lezione 13 pag. 37

Pumping lemma

DEF[`pumping lemma`]

Se A è un linguaggio regolare, allora $\exists p > 0$ (**costante** del pumping) tale che $\forall s \in A$ di lunghezza almeno p (cioè $|s| \geq p$), **esistono** stringhe x, y, z tali che

$$s = xyz$$

per cui valgono le seguenti **condizioni**:

- $xy^iz \in A$, per ogni $i \geq 0$,
- $|y| \geq 1$,
- $|xy| \leq p$.

Lezione 13 pag. 91

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 $i = 2$, $xy^2z = a^{p+m} b^p \notin L$.

Contraddizione

$$\underbrace{a\,a\,a\,a\,a}_{x}\;\underbrace{a\,a}_{y}\;\underbrace{a\,a\,a}_{y}\;\underbrace{a\,a\,b\,b\,b\,b\,b\,b\,b\,b\,b}_{z}$$

Linguaggi regolari: dimostrare o confutare

Dimostrare o confutare le seguenti affermazioni.

(a) Il linguaggio $X = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ è regolare (indichiamo con $|w|_a$ e $|w|_b$ rispettivamente le occorrenze di a e di b in w).

(b) Il linguaggio $Y = \{a^i b^j \mid i + j$ è multiplo di $2\}$ è regolare.

(c) $L((ab)^*) \cap L((cd)^*) = \emptyset$, dove $L(E)$ è il linguaggio denotato dall'espressione regolare E.

Lezione 5 pag. 22

Linguaggio riconosciuto da un DFA: esempi

Linguaggi regolari: dimostrare o confutare

Dimostrare o confutare le seguenti affermazioni.

(a) Il linguaggio $X = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ è regolare (indichiamo con $|w|_a$ e $|w|_b$ rispettivamente le occorrenze di a e di b in w).

(b) Il linguaggio $Y = \{a^i b^j \mid i + j$ è multiplo di $2\}$ è regolare.

(c) $L((ab)^*) \cap L((cd)^*) = \emptyset$, dove $L(E)$ è il linguaggio denotato dall'espressione regolare E.

Lezione 12 pag. 72

Esempio

$E = (a \cup b) \cdot a^*$ L(E)=?

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

$$=$$

$$=$$

$$=$$

$$=$$

Lezione 12 pag. 72

Esempio

$E = (a \cup b) \cdot a^*$ L(E)=?

$$
\begin{aligned}
L((a \cup b) \cdot a^*) &= L(a \cup b) \cdot L(a^*) \\
&= (L(a) \cup L(b)) \cdot (L(a))^* \\
&= (\{a\} \cup \{b\}) \cdot (\{a\})^* \\
&= \{a, b\}\{\epsilon, a, aa, aaa, \dots\} \\
&= \{a, aa, aaa, \dots, b, ba, baa, \dots\}
\end{aligned}
$$

Linguaggi regolari: DFA ed espressione

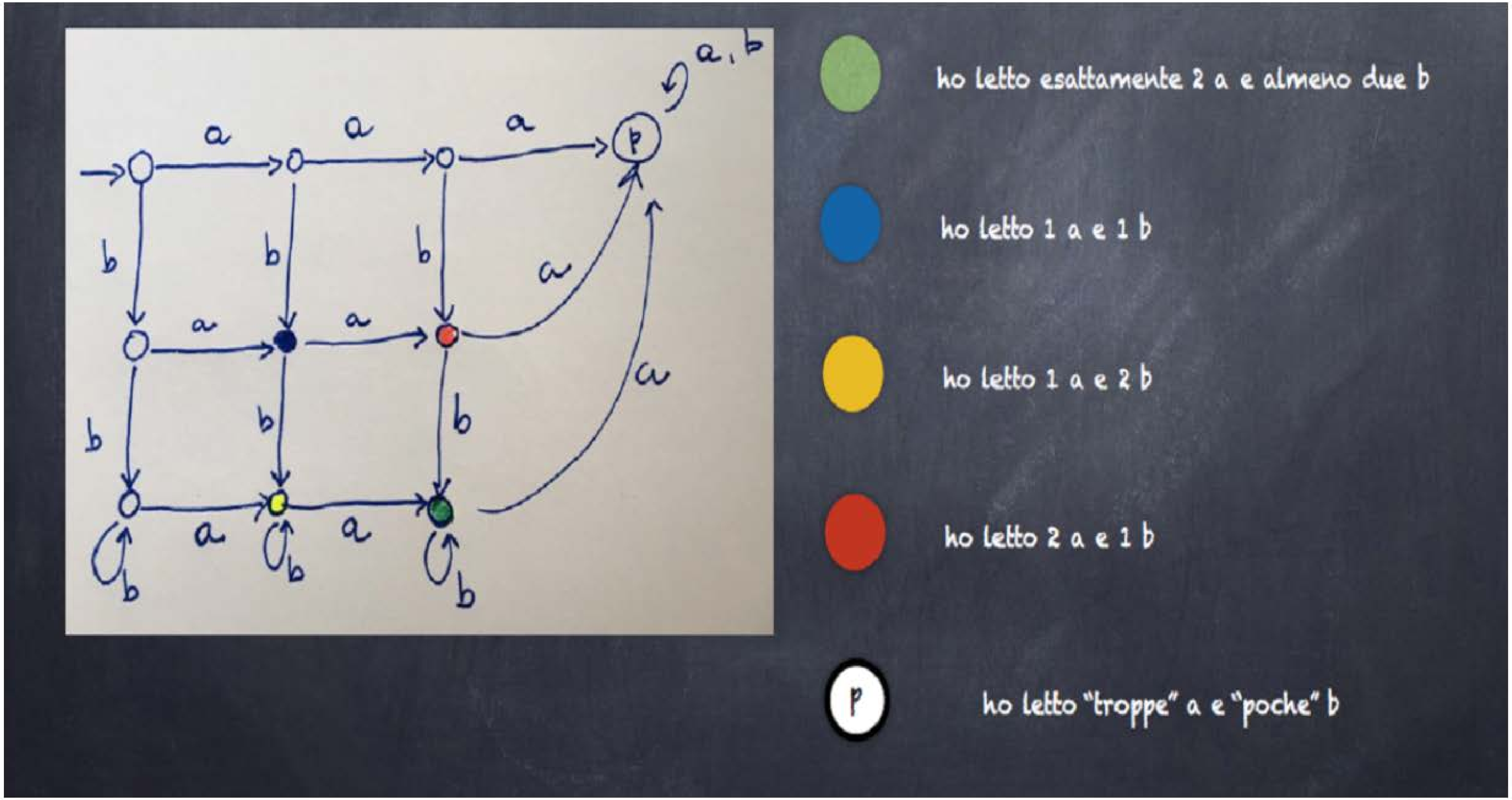
Sia L l'insieme delle stringhe su $\{a, b\}$ che contengono almeno due occorrenze di a e al più una occorrenza di b.

a) Fornire un DFA che riconosce L.

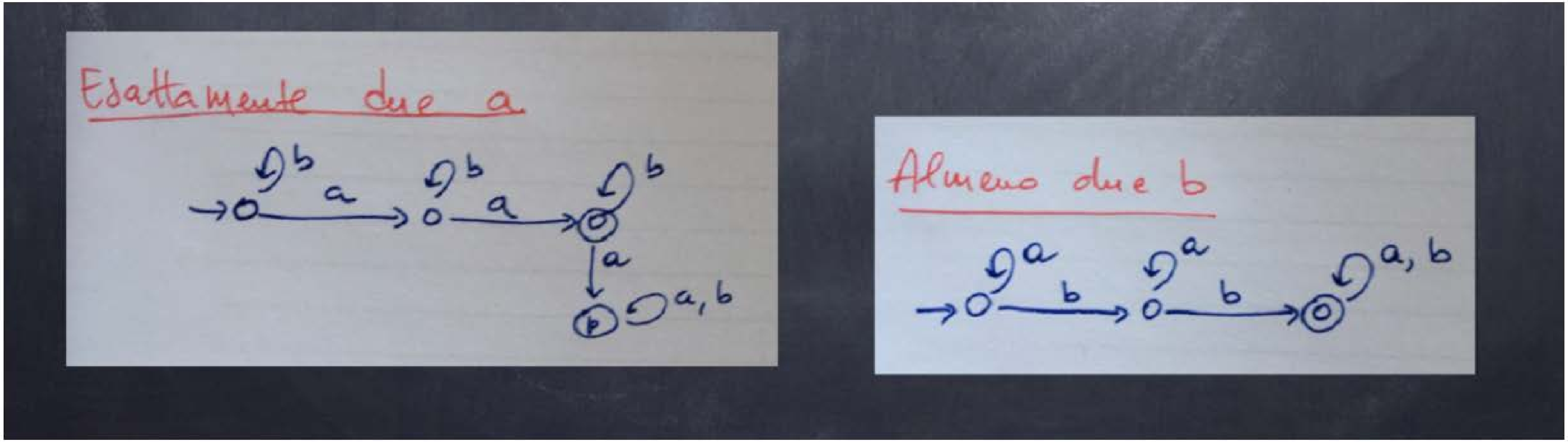
b) Fornire un'espressione regolare che denota L.

Lezione 7 pag. 21

Chiusura di REG rispetto all'intersezione: tecnica 1



Chiusura di REG rispetto all'intersezione: tecnica 2



Costruire l'automa ottenuto dall'intersezione...e confrontare gli automi ottenuti!

Lezione 6 pag. 62

Chiusura di REG rispetto all'~~unione~~ intersezione

Costruzione formale

Siano $\mathbb{M}_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $\mathbb{M}_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

Definiamo $\mathbb{M} = (Q, \Sigma, \delta, q_0, F)$ che riconosce $L_1 \cup L_2$ ($L(\mathbb{M}) = L_1 \cup L_2$), come segue:

- $Q = \{(q_1, q_2) \mid q_1 \in Q_1 \text{ e } q_2 \in Q_2\}$: insieme **prodotto cartesiano** di $Q_1 \times Q_2$.
- Σ è lo stesso alfabeto usato in $\mathbb{M}_1$ e $\mathbb{M}_2$: per semplicità assumiamo che $\mathbb{M}_1$ e $\mathbb{M}_2$ usino lo stesso alfabeto, ma continua ad essere vero anche se hanno alfabeti diversi ($\Sigma = \Sigma_1 \cup \Sigma_2$).
- For each $(q_1, q_2) \in Q$ e ogni $a \in \Sigma$:

$$\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$$

- $q_0 = (q_1, q_2)$
- F is the set of pairs in which either member is an accept state of $\mathbb{M}_1$ or $\mathbb{M}_2$:

$$F = \{(q_1, q_2) \mid q_1 \in F_1 \ \overset{\text{e}}{\cancel{\text{oppure}}}\ q_2 \in F_2\}$$

Linguaggi regolari: DFA ed espressione

Sia L l'insieme delle stringhe su $\{a, b\}$ che contengono almeno due occorrenze di a e al più una occorrenza di b.

a) Fornire un DFA che riconosce L.

b) Fornire un'espressione regolare che denota L.

Lezione 12 pag. 71

Esercizio (1.53, `sipser`)

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

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

Lezione 12 pag. 76

Esempio

$$E = (0 \cup 1)^*00(0 \cup 1)^*$$

L(E)=?

$$L(E) = \{00, 000, 100, 001, 0000, 0001, 1000, 1001, \ldots\}$$

Insieme delle stringhe che contengono 00 come sottostringa

In generale: insieme delle stringhe che contengono X come sottostringa

Lezione 12 pag. 77

Esempio

$E = (1 \cup 01)^*(0 \cup \epsilon)$

L(E)=?

$L(E) = \{\epsilon, 0, 10, 1, 010, 01, 110, 11, 01010, 0101, 1111, 111 \ldots\}$

Insieme delle stringhe che non contengono 00 come sottostringa

Teorema di Rice: dimostrare o confutare

(a) Enunciare il teorema di Rice.

(b) A quale dei seguenti linguaggi è possibile applicarlo?

$$\begin{aligned} X &= \{\langle M \rangle \mid M \text{ è una MdT che rifiuta } ab\} \\ Y &= \{\langle M \rangle \mid M \text{ è una MdT che accetta } ab\} \end{aligned}$$

Lezione 28 pag. 45

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.

Linguaggio Turing riconoscibile e decidibile

(a) Fornire le definizioni di linguaggio, linguaggio Turing riconoscibile, linguaggio decidibile.

(b) Dimostrare o confutare le seguenti affermazioni, giustificando la risposta. Occorre enunciare con precisione gli eventuali risultati intermedi utilizzati. È possibile limitarsi a una descrizione ad alto livello delle macchine di Turing utilizzate.

- La classe dei linguaggi decidibili è chiusa rispetto al complemento.
- La classe dei linguaggi Turing riconoscibili è chiusa rispetto al complemento.

Lezione 03 pag. 73

Linguaggi

DEF[`linguaggio formale`]

Un linguaggio formale è un **insieme di stringhe** su un alfabeto.

L è un linguaggio sull'alfabeto Σ se $L \subseteq \Sigma^*$

Può essere infinito!

Esempi..

Sia $\Sigma = \{a\}$ un alfabeto. Consideriamo $L = \{\epsilon, a, aaa, aaaaa, \ldots\} = \{a^{2n+1} \mid n \in \mathbb{N}, n \geq 0\}$.

Poichè $L \subseteq \Sigma^*$, allora L è un linguaggio su Σ.

Lezione 17 pag. 31

Dal punto di vista dei linguaggi

Definizione

Un linguaggio $L \subseteq \Sigma^*$ è Turing riconoscibile se esiste una macchina di Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ tale che:

1. *M riconosce L (cioè $L = L(M) = \{w \in \Sigma^* \mid \exists u, v \in \Gamma^* \; q_0 w \to^* u q_{accept} v\}$).*

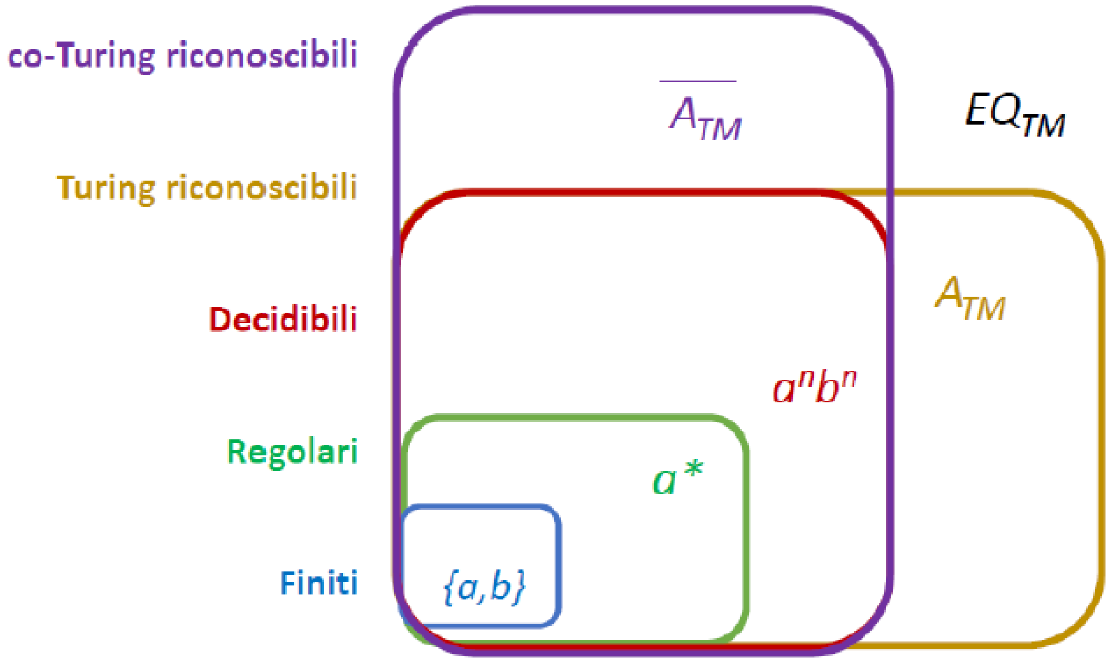
Definizione

Un linguaggio $L \subseteq \Sigma^*$ è decidibile se esiste una macchina di Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ tale che:

1. *M riconosce L (cioè $L = L(M) = \{w \in \Sigma^* \mid \exists u, v \in \Gamma^* \; q_0 w \to^* u q_{accept} v\}$).*
2. *M si arresta su ogni input (cioè per ogni $w \in \Sigma^*$, $q_0 w \to^* uqv$ con $q \in \{q_{accept}, q_{reject}\}$).*

Lezione 28 pag. 22

The right picture



Decidibili = Turing riconoscibili ∩ co-Turing riconoscibili

Lezione 26 pag. 85

Esercizio 1

La classe dei linguaggi **decidibili** è chiusa rispetto al complemento?

Soluzione:

La classe dei linguaggi decidibili è chiusa rispetto al complemento. Sia A un linguaggio decidibile, sia M_A una macchina di Turing che decide A.

Definiamo la macchina di Turing $M_{\overline{A}}$: sull'input w, $M_{\overline{A}}$ simula M_A e accetta w se e solo se M_A rifiuta w.

Poiché M_A si arresta su ogni input anche $M_{\overline{A}}$ si arresta su ogni input.

Inoltre, il linguaggio di $M_{\overline{A}}$ è $\overline{A}$ perché $M_{\overline{A}}$ accetta w se e solo se M_A rifiuta w e quindi se e solo se $w \notin A$.

Quindi $M_{\overline{A}}$ è una macchina di Turing che decide $\overline{A}$ e $\overline{A}$ è decidibile.

Lezione 26 pag. 86

Soluzione formale

Formalmente, se

$$M_A = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$$

definiamo

$$M_{\overline{A}} = (Q, \Sigma, \Gamma, \delta', q_0, q_{accept}, q_{reject})$$

dove, per ogni $q \in Q \setminus \{q_{accept}, q_{reject}\}$, per ogni $\gamma \in \Gamma$

$$\delta'(q, \gamma) = \begin{cases} \delta(q, \gamma) & \text{se } \delta(q, \gamma) = (q', \gamma', d), \\ & \text{con } q' \notin \{q_{accept}, q_{reject}\}, \\ (q_{accept}, \gamma', d) & \text{se } \delta(q, \gamma) = (q_{reject}, \gamma', d), \\ (q_{reject}, \gamma', d) & \text{se } \delta(q, \gamma) = (q_{accept}, \gamma', d) \end{cases}$$

Figura:

Lezione 26 pag. 87

Esercizio 2

La classe dei linguaggi **riconoscibili** è chiusa rispetto al complemento?

Lezione 26 pag. 83

Un linguaggio che non è Turing riconoscibile

Teorema

$\overline{A_{TM}}$ non è Turing riconoscibile.

Dimostrazione.

Supponiamo per assurdo che $\overline{A_{TM}}$ sia Turing riconoscibile.
Sappiamo che A_{TM} è Turing riconoscibile.
Quindi A_{TM} è Turing riconoscibile e co-Turing riconoscibile.
Per il precedente teorema, A_{TM} è decidibile.
Assurdo, poichè abbiamo dimostrato che A_{TM} è indecidibile.

$\square$

Lezione 26 pag. 55

A_{TM} è Turing riconoscibile

Teorema

Il linguaggio

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT che accetta la parola } w\}$$

è Turing riconoscibile.

Lezione 26 pag. 56

Dimostrazione

La seguente macchina di Turing U riconosce A_{TM}.

$U =$ "Sull'input $\langle M, w \rangle$ dove M è una TM e w è una stringa

1. Simula M sull'input w.
2. Se M accetta w, accetta l'input $\langle M, w \rangle$; se M rifiuta w, rifiuta l'input $\langle M, w \rangle$."

U rifiuta ogni stringa che non sia della forma $\langle M, w \rangle$ dove M è una TM e w è una stringa.

Quindi U accetta una stringa y se e solo se y è della forma $\langle M, w \rangle$ dove M è una TM, w è una stringa e M accetta w.

In altri termini, U accetta una stringa y se e solo se $y = \langle M, w \rangle$ è un elemento di A_{TM}.

Ne segue $L(U) = A_{TM}$.

Non richiesta dalla traccia, solo di riepilogo

Lezione 26 pag. 36

Un problema indecidibile

$$A_{TM} = \{\langle M, w \rangle \mid M \text{ è una MdT e } M \text{ accetta } w\}$$

A_{TM} è il linguaggio associato al problema decisionale dell'accettazione di una macchina di Turing.

Teorema

Il linguaggio A_{TM} non è decidibile.

Lezione 26 pag. 46

A_{TM} è indecidibile: riepilogo della dimostrazione

1. Definiamo $A_{TM} = \{\langle M, w\rangle \mid M \text{ è MdT che accetta } w\}$
2. Assumiamo A_{TM} decidibile; sia H MdT che lo decide
3. Usiamo H per costruire MdT D che inverte le decisioni;
   $D(\langle M\rangle)$: accetta se M non accetta $\langle M\rangle$; rifiuta se M accetta $\langle M\rangle$.
4. Diamo in input a D la sua codifica $\langle D\rangle$:
   $D(\langle D\rangle)$ accetta sse D rifiuta.

Contraddizione

Non richiesta
dalla traccia,
solo di riepilogo

Immagine nella riduzione da 3SAT a VERTEX-COVER

Data la seguente formula booleana

$$\Phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$$

definire il grafo G e l'intero k tali che $\langle G, k \rangle$ sia l'immagine di $\langle \Phi \rangle$ nella riduzione polinomiale di 3-SAT a $VERTEX$-$COVER$.

Lezione 33-34(a) pag. 29

$3SAT$ si riduce in tempo polinomiale a $VERTEX\text{-}COVER$

Costruzione:

- G contiene due vertici per ogni variabile x etichettati con x e $\overline{x}$ (gadget per le variabili). Chiamiamo V_1 questo insieme di vertici.
- G contiene tre vertici per ogni clausola, etichettati con i tre letterali della clausola (gadget per le clausole). Chiamiamo V_2 questo insieme di vertici.
- Connettiamo i due vertici associati a una variabile con un arco
- Connettiamo i tre vertici associati a una clausola tra loro in un triangolo
- Connettiamo con un arco ogni vertice nel triangolo (associato a una clausola) al vertice in V_1 (gadget per le variabili) che ha la stessa etichetta.
- Se ϕ ha ℓ clausole e m variabili allora G ha $2m + 3\ell$ vertici e $V = V_1 \cup V_2$.
- Prendiamo $k = m + 2\ell$.

Lezione 33-34(a) pag. 36

$3SAT$ si riduce in tempo polinomiale a $VERTEX\text{-}COVER$

Proviamo che ϕ è soddisfacibile se e solo se $G = (V, E)$ ha un vertex cover di cardinalità k.

- Sia ϕ soddisfacibile e sia τ un assegnamento che soddisfa ϕ. Consideriamo il sottoinsieme V' di V che contiene:
  - tutti i vertici in V_1 (gadget per le variabili) che hanno come etichette i letterali veri in τ
  - due vertici per ogni triangolo (gadget per una clausola), escludendone uno che ha etichetta uguale a un vertice selezionato al passo precedente (ne esiste almeno uno).
- Se ϕ ha ℓ clausole e m variabili allora questo sottoinsieme V' di V ha taglia $k = m + 2\ell$.
- Inoltre V' è un vertex cover:
  - tutti gli archi in un triangolo sono coperti (dai due vertici selezionati)
  - tutti gli archi tra due vertici di V_1 o tra un vertice di V_1 e un vertice di V_2 sono coperti (dalla scelta dei vertici in V_1 o V_2).

Per questo esercizio basta fermarsi qui, le slide successive sono di riepilogo

Lezione 33-34(a) pag. 39

$3SAT$ si riduce in tempo polinomiale a $VERTEX\text{-}COVER$

- Supponiamo che $G = (V, E)$ abbia un vertex cover V' di cardinalità $k = m + 2\ell$ e proviamo che ϕ è soddisfacibile.
- V' deve contenere almeno 2 vertici di ogni triangolo e, per ogni arco tra due vertici di V_1 (gadget per le variabili), almeno 1 dei 2 vertici.
- Siccome il numero dei triangoli è ℓ e il numero degli archi tra i vertici in V_1 è m, l'insieme V' contiene **esattamente** due vertici di ogni triangolo e, per ogni arco tra due vertici di V_1, uno dei 2 vertici.

Lezione 33-34(a) pag. 45

3*SAT* si riduce in tempo polinomiale a *VERTEX*-*COVER*

- Assegniamo valore vero ai letterali che sono etichette di vertici in $V' \cap V_1$.
- Proviamo che questo assegnamento τ soddisfa ϕ. Cioè che questo assegnamento rende vera ogni clausola.
- Infatti, per ogni triangolo esiste un vertice u che non è in V'.
- Ma (u, v), con $v \in V_1$ deve essere coperto da V'. Quindi $v \in V'$.
- Ma u e v hanno la stessa etichetta (per costruzione di G) che corrisponde a un letterale a cui è assegnato valore 1 (per costruzione di τ).
- Dunque, per ogni clausola c, c'è un letterale a cui τ assegna valore 1 e quindi ϕ è soddisfacibile. $\square$

Prossimo tutorato

Prima del secondo appello di luglio:
data da definire, la troverete pianificata su questo canale del Team…
…buono studio
e in bocca al lupo
per l'appello
di domani ☺