Elementi di Teoria della Computazione

Classe: Resto_2 - Prof.ssa Marcella Anselmo

Tutorato

18/07/2022 ore 11:00-13:00

Ottava Esercitazione

a cura della dott.ssa Manuela Flores

Appello 05/07/2022: linguaggi regolari

1. Scrivere se le seguenti affermazioni sono vere o false, giustificando la risposta.

   (a) $X = \{a^k baba^k \mid k \leq 5\}$ è regolare.

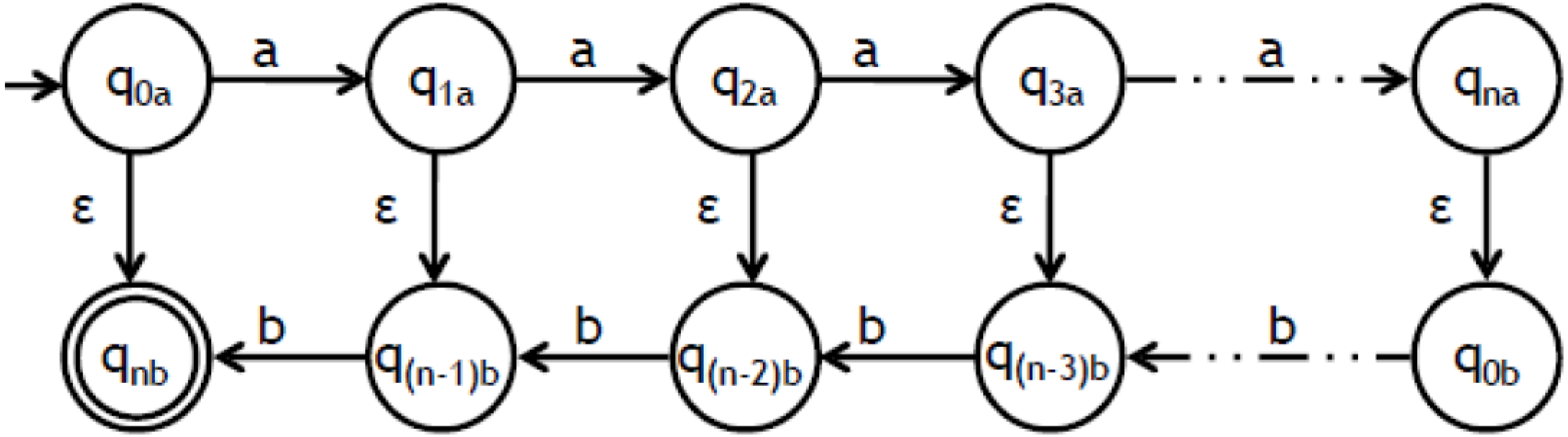
   (b) $Y = \{a^i b^j \mid i < j\}$ è regolare.

   (c) Ogni sottoinsieme proprio di un linguaggio regolare è regolare.

Lezione 13 pag. 23

Linguaggi regolari vs. non regolari

Consideriamo il linguaggio $L_n = \{a^k b^k \mid k \leq n\}$, $\forall n \geq 0$. **E' regolare o no?**



L'NFA costruito riconosce L_n, quindi L_n è regolare!

Appello 05/07/2022: linguaggi regolari

1. Scrivere se le seguenti affermazioni sono vere o false, giustificando la risposta.

   (a) $X = \{a^k baba^k \mid k \leq 5\}$ è regolare.

   (b) $Y = \{a^i b^j \mid i < j\}$ è regolare.

   (c) Ogni sottoinsieme proprio di un linguaggio regolare è regolare.

(a)

Appello 05/07/2022: linguaggi regolari

1. Scrivere se le seguenti affermazioni sono vere o false, giustificando la risposta.

   (a) $X = \{a^k baba^k \mid k \leq 5\}$ è regolare.

   (b) $Y = \{a^i b^j \mid i < j\}$ è regolare.

   (c) Ogni sottoinsieme proprio di un linguaggio regolare è regolare.

(b)

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 $|y| \geq 1$.

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

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

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

Appello 05/07/2022: linguaggi regolari

1. Scrivere se le seguenti affermazioni sono vere o false, giustificando la risposta.

   (a) $X = \{a^k baba^k \mid k \leq 5\}$ è regolare.

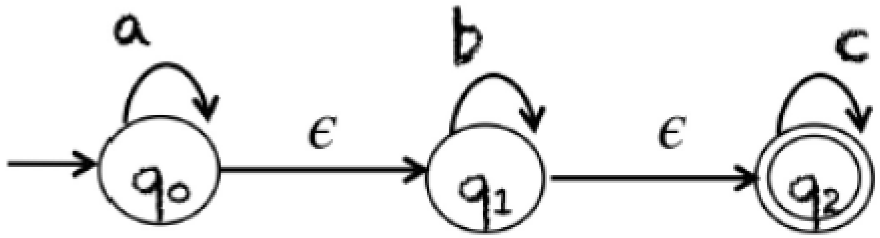
   (b) $Y = \{a^i b^j \mid i < j\}$ è regolare.

   (c) Ogni sottoinsieme proprio di un linguaggio regolare è regolare.

(c)

Appello 05/07/2022: da NFA a DFA

2. Trasformare il seguente NFA nel DFA equivalente utilizzando la costruzione presentata nella dimostrazione del Teorema sull'equivalenza NFA-DFA. 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 accettato dall'automa.



Subset construction: da NFA a DFA

Costruzione.

Sia $\mathbb{N} = (Q_N, \Sigma, \delta_N, q_N, F_N)$ un NFA, costruiamo il DFA $\mathbb{M} = (Q_M, \Sigma, \delta_M, q_M, F_M)$ come:

1. $Q_M = P(Q_N)$, **insieme potenza** di Q_N; osserviamo che $|P(Q_N)| = 2^{|Q_N|}$.
   - Q_M contiene tutti i "possibili stati" in cui può terminare una transizione di $\mathbb{M}$, cioè tutte le combinazioni possibili di stati di Q_N.
2. $q_M = E(q_N)$, lo stato iniziale di $\mathbb{M}$ non è solo q_N ma anche tutti gli stati raggiungibili da q_N utilizzando solo ϵ-transizioni, quindi $E(q_N)$.
3. $F_M = \{R \in Q_M \mid R \cap F_N \neq \emptyset\}$
4. $\forall R \in Q_M$, $\forall a \in \Sigma$:

$$\delta_M(R, a) = E(\cup_{r \in R} \delta_N(r, a)) = \cup_{r \in R} E(\delta_N(r, a))$$

NFA: computazione (HUM, 2.5.3–2.5.4)

Come per i DFA, siamo interessati a definire le computazioni di δ in termini di stringhe.

Innanzitutto, definiamo l'insieme degli stati raggiungibili da uno stato usando solo ϵ-transizioni.

Sia $\mathbb{M} = (Q, \Sigma, \delta, q_0, F)$ un **NFA** e $q \in Q$. La **ϵ-chiusura** $E(q)$ di q è il sottinsieme di Q definito ricorsivamente come segue:

passo base: $q \in E(q)$

passo ricorsivo: $\forall p \in E(q)$, $\delta(p, \epsilon) \subseteq E(q)$

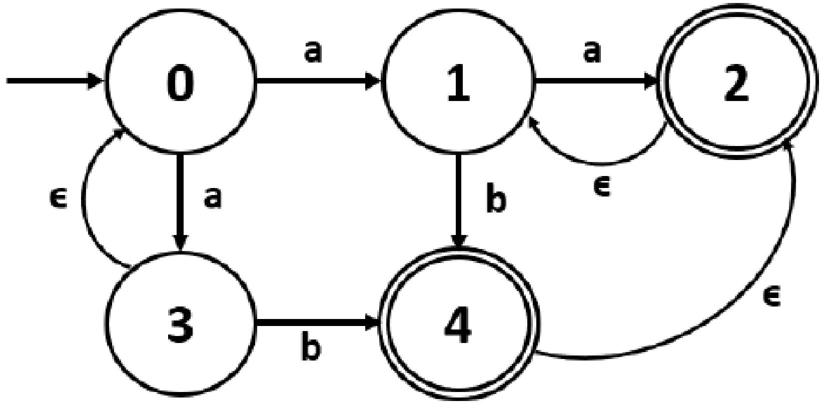
Sia $R \subseteq Q$. La ϵ-chiusura $E(R)$ di R è:

$$E(R) = \cup_{q \in R} E(q)$$

Lezione 9 pag. 98

Subset construction: da NFA a DFA (esempio)

Consideriamo il seguente NFA $\mathbb{N} = (Q_N, \Sigma, \delta_N, 0, F_N)$

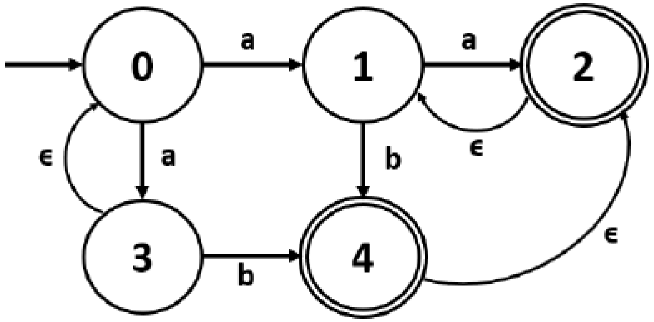


	a	b	ϵ
0	{1,3}	∅	∅
1	{2}	{4}	∅
2	∅	∅	{1}
3	∅	{4}	{0}
4	∅	∅	{2}

Costruiamo innanzitutto le ϵ-chiusure: $E(0) = \{0\}$, $E(1) = \{1\}$, $E(2) = \{2, 1\}$, $E(3) = \{3, 0\}$, e $E(4) = \{4, 2, 1\}$.

Lezione 9 pag. 111

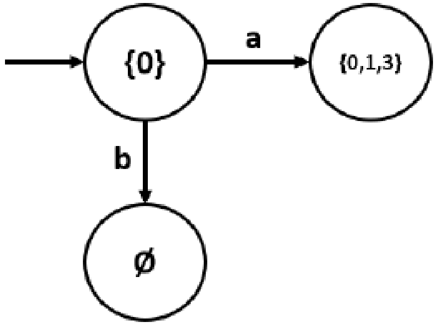
Subset construction: da NFA a DFA (esempio)



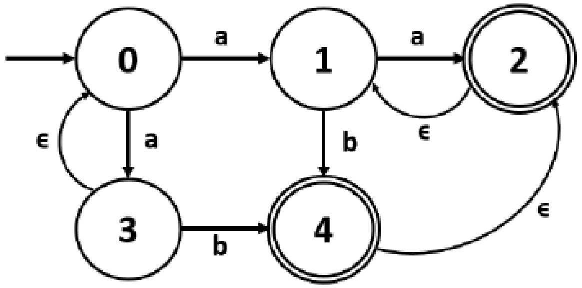
	a	b	ϵ
0	{1,3}	∅	∅
1	{2}	{4}	∅
2	∅	∅	{1}
3	∅	{4}	{0}
4	∅	∅	{2}

Costruiamo le transizioni per ogni stato:

- $\delta_M(\{0\}, a) = E(\delta_N(0, a)) = E(\{1, 3\}) = E(\{1\}) \cup E(\{3\}) = \{0, 1, 3\}$
- $\delta_M(\{0\}, b) = E(\delta_N(0, b)) = E(\emptyset) = \emptyset$



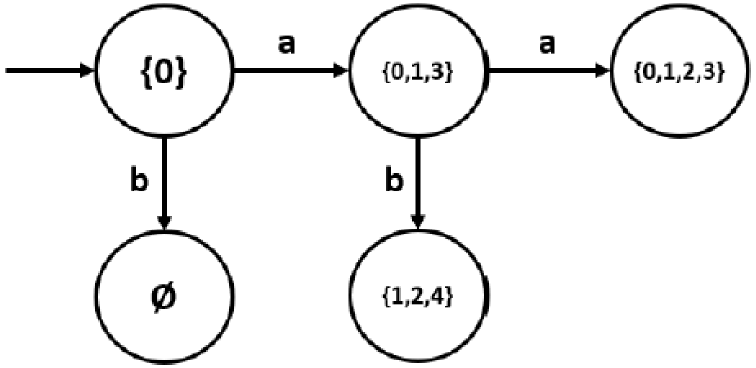
Subset construction: da NFA a DFA (esempio)



	a	b	ϵ
0	{1,3}	∅	∅
1	{2}	{4}	∅
2	∅	∅	{1}
3	∅	{4}	{0}
4	∅	∅	{2}

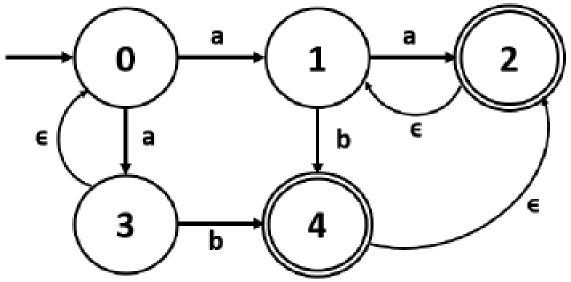
Costruiamo le transizioni per ogni stato:

- $\delta_M(\{0,1,3\},a) = E(\delta_N(0,a) \cup \delta_N(1,a) \cup \delta_N(3,a)) = E(\{1,3\} \cup \{2\} \cup \emptyset) = \{0,1,2,3\}$
- $\delta_M(\{0,1,3\},b) = E(\delta_N(0,b) \cup \delta_N(1,b) \cup \delta_N(3,b)) = E(\emptyset \cup \{4\} \cup \{4\}) = E(\{4\}) = \{1,2,4\}$



Lezione 9 pag. 122

Subset construction: da NFA a DFA (esempio)



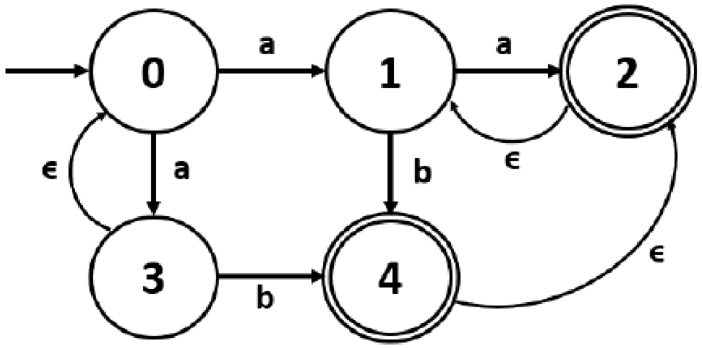
	a	b	ϵ
0	{1,3}	∅	∅
1	{2}	{4}	∅
2	∅	∅	{1}
3	∅	{4}	{0}
4	∅	∅	{2}

Costruiamo le transizioni per ogni stato:

- $\delta_M(\{1,2,4\},a) = E(\{2\} \cup \emptyset \cup \emptyset) = \{1,2\}$
- $\delta_M(\{1,2,4\},b) = E(\{4\} \cup \emptyset \cup \emptyset) = \{1,2,4\}$
- $\delta_M(\{0,1,2,3\},a) = E(\{1,3\} \cup \{2\} \cup \emptyset \cup \emptyset) = \{1,3,0,2\}$
- $\delta_M(\{0,1,2,3\},b) = E(\emptyset \cup \{4\} \cup \emptyset \cup \{4\}) = \{1,2,4\}$
- $\delta_M(\{1,2\},a) = E(\{2\} \cup \emptyset) = \{1,2\}$
- $\delta_M(\{1,2\},b) = E(\{4\} \cup \emptyset) = \{1,2,4\}$

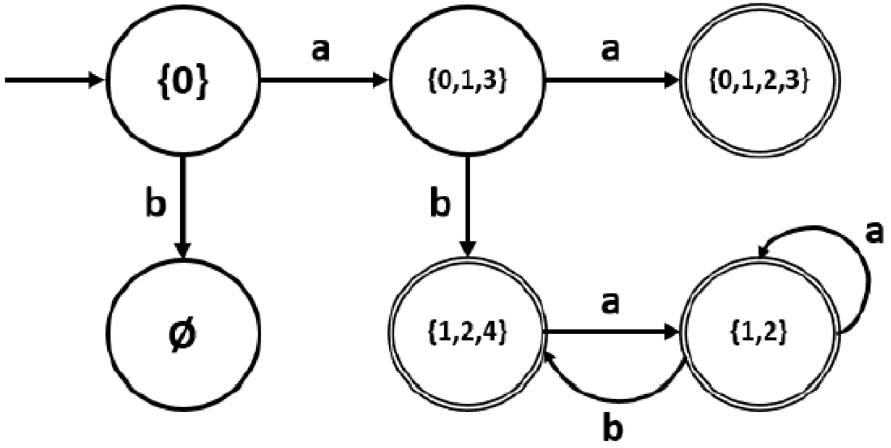
Lezione 9 pag. 123

Subset construction: da NFA a DFA (esempio)



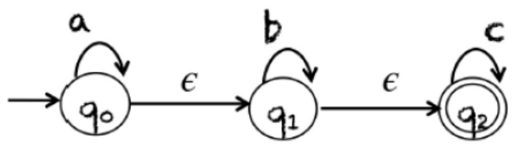
	a	b	ϵ
0	{1,3}	∅	∅
1	{2}	{4}	∅
2	∅	∅	{1}
3	∅	{4}	{0}
4	∅	∅	{2}

Consideriamo solo le **parti raggiungibili** dallo stato iniziale del DFA.



Appello 05/07/2022: da NFA a DFA

2. Trasformare il seguente NFA nel DFA equivalente utilizzando la costruzione presentata nella dimostrazione del Teorema sull'equivalenza NFA-DFA. 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 accettato dall'automa.



Appello 05/07/2022: Computazione di MdT

Esercizio 3 (*5 punti*)

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

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

a) Indicare (se esistono)
- una stringa $\mathbf{w_a}$ di Σ^* che sia **accettata** da M con la relativa **computazione**
- una stringa $\mathbf{w_r}$ di Σ^* che sia **rifiutata** da M con la relativa **computazione**
- una stringa $\mathbf{w_c}$ di Σ^* su cui M **cicla**

b) Descrivere il linguaggio L(M) **riconosciuto** da M.

c) Il linguaggio L(M) è anche **deciso** da M? Motivare pienamente la risposta.

Lezione 16 pag. 36

Computazione di una MdT

Siano C, C' *configurazioni.*
$C \to^* C'$ *se esistono configurazioni* $C_1, \ldots, C_k$, $k \geq 1$ *tali che*

1. $C_1 = C$,
2. $C_i \to C_{i+1}$, *per* $i \in \{1, \ldots, k-1\}$,
   (ogni C_i *produce* C_{i+1}*)*
3. $C_k = C'$.

Diremo che $C \to^* C'$ *è una* **computazione** *(di lunghezza* $k-1$*).*

Quando k = 1?

Lezione 16 pag. 37

Configurazioni

Una configurazione C si dice:

- iniziale su input w se $C = q_0 w$, con w $\in \Sigma^*$
- di accettazione se $C = u\ q_{accept}\ v$
- di rifiuto se $C = u\ q_{reject}\ v$

Poiché non esistono transizioni da q_{accept} e da q_{reject}, allora le configurazioni di accettazione e di rifiuto sono dette configurazioni di arresto.

Lezione 16 pag. 35

Esempio

$$\begin{aligned}
\delta(q_0, 0) &= (q_0, 0, R), \quad \delta(q_0, 1) = (q_0, 1, R),\\
\delta(q_0, \sqcup) &= (q_1, \sqcup, L),\\
\delta(q_1, 1) &= (q_2, 1, L), \quad \delta(q_2, 0) = (q_3, 0, L),\\
\delta(q_3, 1) &= (q_{accept}, 1, L)
\end{aligned}$$

$q_0 11 \to 1 q_0 1 \to 11 q_0 \to 1 q_1 1 \to q_2 11 \to q_{reject} 11$

$q_0 101 \to 1 q_0 01 \to 10 q_0 1 \to 101 q_0 \to 10 q_1 1 \to 1 q_2 01 \to$
$q_3 101 \to q_{accept} 101$

Appello 05/07/2022: Computazione di MdT

Esercizio 3 (*5 punti*)

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

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

a) Indicare (se esistono)
   - una stringa $\mathbf{w_a}$ di Σ^* che sia **accettata** da M con la relativa **computazione**
   - una stringa $\mathbf{w_r}$ di Σ^* che sia **rifiutata** da M con la relativa **computazione**
   - una stringa $\mathbf{w_c}$ di Σ^* su cui M **cicla**

b) Descrivere il linguaggio L(M) **riconosciuto** da M.

c) Il linguaggio L(M) è anche **deciso** da M? Motivare pienamente la risposta.

Appello 05/07/2022: EQ_TM e ¬EQ_TM

Esercizio 4 (*3 punti*)

a) **Definire** il linguaggio **EQ$_{\mathbf{TM}}$**.

b) Provare che il **complemento di EQ$_{\mathbf{TM}}$** non è Turing-riconoscibile.
Enunciare con precisione eventuali risultati noti che vengono utilizzati, senza necessariamente dimostrarli.

Lezione 28 pag. 25

EQ_{TM} è indecidibile

$E_{TM} = \{\langle M \rangle \mid M \text{ è una } MdT \text{ e } L(M) = \emptyset\}$
$EQ_{TM} = \{\langle M_1, M_2 \rangle \mid M_1, M_2 \text{ sono MdT e } L(M_1) = L(M_2)\}$

$$E_{TM} \leq_m EQ_{TM}$$

Sia M_1 una macchina di Turing tale che $L(M_1) = \emptyset$.
$f : \langle M \rangle \to \langle M, M_1 \rangle$ è una riduzione di E_{TM} a EQ_{TM}.
Perchè?

Lezione 28 pag. 29

Riduzione da A_{TM} a EQ_{TM}

$$A_{TM} \leq_m EQ_{TM}$$

Idea: Data $\langle M, w \rangle$, considerare le MdT M_1 e M_2 tali che

Per ogni input x:

M_1 accetta x,

M_2 simula M su w. Se M accetta w, M_2 accetta x.

$f : \langle M, w \rangle \to \langle M_1, M_2 \rangle$ è riduzione da A_{TM} a EQ_{TM}.

Perchè?

$$L(M_1) = \Sigma^*; \; L(M_2) = \begin{cases} \Sigma^* & \text{se } \langle M, w \rangle \in A_{TM} \\ \emptyset & \text{se } \langle M, w \rangle \notin A_{TM} \end{cases}$$

Lezione 28 pag. 32

Riduzione da A_{TM} al complemento di EQ_{TM}

$f : \langle M, w\rangle \rightarrow \langle M_1, M_2\rangle$ è riduzione che prova $A_{TM} \leq_m EQ_{TM}$.

$$L(M_1) = \Sigma^*;\ L(M_2) = \begin{cases} \Sigma^* & \text{se } \langle M, w\rangle \in A_{TM} \\ \emptyset & \text{se } \langle M, w\rangle \notin A_{TM} \end{cases}$$

Possiamo modificare f per dimostrare che $A_{TM} \leq_m \overline{EQ_{TM}}$?

Lasciamo la stessa M_2 e cambiamo M_1 in $\mathbf{M_3}$.
$g : \langle M, w\rangle \rightarrow \langle M_3, M_2\rangle$

$$\mathbf{L(M_3)} = \emptyset;\ L(M_2) = \begin{cases} \Sigma^* & \text{se } \langle M, w\rangle \in A_{TM} \\ \emptyset & \text{se } \langle M, w\rangle \notin A_{TM} \end{cases}$$

Lezione 27 pag. 19

Teoremi

Teorema

$A \leq_m B$ *se e solo se* $\overline{A} \leq_m \overline{B}$.

Dimostrazione

Per ipotesi $A \leq_m B$, quindi esiste una riduzione di A a B.

Poiché f è una riduzione, f è calcolabile e inoltre

$$\forall w \in \Sigma^* \quad w \in A \Leftrightarrow f(w) \in B$$

Proviamo che f è anche una riduzione da $\overline{A}$ a $\overline{B}$.

Lezione 26 pag. 43

Una proprietà dei linguaggi decidibili

Definizione
Diciamo che un linguaggio L è co-Turing riconoscibile se $\overline{L}$ è Turing riconoscibile.

Teorema
Un linguaggio L è decidibile se e solo se L è Turing riconoscibile **e** *co-Turing riconoscibile.*

Lezione 28 pag. 37

Linguaggi riconoscibili e co-Turing riconoscibili

Teorema

EQ_{TM} non è nè Turing riconoscibile nè co-Turing riconoscibile.

Dimostrazione.

Supponiamo per assurdo che EQ_{TM} sia Turing riconoscibile.

$A_{TM} \leq_m \overline{EQ_{TM}} \Rightarrow \overline{A_{TM}} \leq_m EQ_{TM}$

Quindi $\overline{A_{TM}}$ sarebbe Turing riconoscibile: assurdo.

Supponiamo per assurdo che EQ_{TM} sia co-Turing riconoscibile, cioè che $\overline{EQ_{TM}}$ sia Turing riconoscibile.

$A_{TM} \leq_m EQ_{TM} \Rightarrow \overline{A_{TM}} \leq_m \overline{EQ_{TM}}$

Quindi $\overline{A_{TM}}$ sarebbe Turing riconoscibile: assurdo.

Appello 05/07/2022: INDIPENDENT-SET

Esercizio 5 (*7 punti*)

Sia G = (V, E) un grafo non orientato e sia $I \subseteq V$. Diciamo che I è un **insieme indipendente** in G se nessuna coppia di nodi in I è connessa da un arco. Formalmente, per ogni u, v $\in$ I si ha (u,v) $\notin$ E.
Il problema di decisione **INDEPENDENT-SET** è il seguente: Dato un grafo non orientato **G** = (V, E) e un intero positivo **k**, **esiste** un insieme indipendente I in G di **cardinalità k**?

a) Definire il **linguaggio** INDSET associato.
b) Mostrare che **INDSET** appartiene a **NP**.
c) Definire il linguaggio **CLIQUE**.
d) Dimostrare che **CLIQUE** $\leq_p$ **INDSET**, fornendo una opportuna **funzione di riduzione**.
e) Cosa possiamo **dedurre** per INDSET dalle affermazioni b) e d)?

Appello 05/07/2022: INDIPENDENT-SET

Esercizio 5 (*7 punti*)

Sia G = (V, E) un grafo non orientato e sia I $\subseteq$ V. Diciamo che I è un **insieme indipendente** in G se nessuna coppia di nodi in I è connessa da un arco. Formalmente, per ogni u, v $\in$ I si ha (u,v) $\notin$ E.
Il problema di decisione **INDEPENDENT-SET** è il seguente: Dato un grafo non orientato **G** = (V, E) e un intero positivo **k**, **esiste** un insieme indipendente I in G di **cardinalità k**?

a) Definire il **linguaggio** INDSET associato.
b) Mostrare che **INDSET** appartiene a **NP**.
c) Definire il linguaggio **CLIQUE**.
d) Dimostrare che **CLIQUE** $\leq_p$ **INDSET**, fornendo una opportuna **funzione di riduzione**.
e) Cosa possiamo **dedurre** per INDSET dalle affermazioni b) e d)?

a)

Appello 05/07/2022: INDIPENDENT-SET

Esercizio 5 (*7 punti*)

Sia G = (V, E) un grafo non orientato e sia I $\subseteq$ V. Diciamo che I è un **insieme indipendente** in G se nessuna coppia di nodi in I è connessa da un arco. Formalmente, per ogni u, v $\in$ I si ha (u,v) $\notin$ E.
Il problema di decisione **INDEPENDENT-SET** è il seguente: Dato un grafo non orientato **G** = (V, E) e un intero positivo **k**, **esiste** un insieme indipendente I in G di **cardinalità k**?

a) Definire il **linguaggio** INDSET associato.
b) Mostrare che **INDSET** appartiene a **NP**.
c) Definire il linguaggio **CLIQUE**.
d) Dimostrare che **CLIQUE** $\leq_p$ **INDSET**, fornendo una opportuna **funzione di riduzione**.
e) Cosa possiamo **dedurre** per INDSET dalle affermazioni b) e d)?

b)

Lezione 30 pag. 19

Esempi di linguaggi in NP

Teorema
$CLIQUE \in NP$

Dimostrazione.
Un algoritmo V che verifica $CLIQUE$ in tempo polinomiale:
$V =$ "Sull'input $\langle\langle G, k\rangle, c\rangle$:

1. Verifica se c è un insieme di k nodi di G, altrimenti rifiuta.
2. Verifica se per ogni coppia di nodi in c, esiste un arco in G che li connette, accetta in caso affermativo; altrimenti rifiuta."

$\exists c : \langle\langle G, k\rangle, c\rangle \in L(V) \Leftrightarrow \langle G, k\rangle \in CLIQUE$ $\square$

Prova alternativa: utilizzare le macchine di Turing non deterministiche.

Lezione 32 pag. 5

3SAT e *CLIQUE*

Teorema

$$\boxed{3SAT \leq_P CLIQUE}$$

$3SAT = \{\langle \phi \rangle \mid \phi$ è una formula 3CNF soddisfacibile$\}$

Una formula 3CNF è un *AND* di clausole e tutte le clausole hanno tre letterali.

c)

$CLIQUE =$
$\{\langle G, k \rangle \mid G$ è un grafo non orientato in cui esiste una k-clique$\}$

Ricorda:

Una clique (o cricca) in un grafo non orientato G è un sottografo G' di G in cui ogni coppia di vertici è connessa da un arco.

Una k-clique è una clique che contiene k vertici.

Appello 05/07/2022: INDIPENDENT-SET

Esercizio 5 (*7 punti*)

Sia G = (V, E) un grafo non orientato e sia $I \subseteq V$. Diciamo che I è un **insieme indipendente** in G se nessuna coppia di nodi in I è connessa da un arco. Formalmente, per ogni $u, v \in I$ si ha $(u,v) \notin E$.
Il problema di decisione **INDEPENDENT-SET** è il seguente: Dato un grafo non orientato **G** = (V, E) e un intero positivo **k**, **esiste** un insieme indipendente I in G di **cardinalità k**?

a) Definire il **linguaggio** INDSET associato.
b) Mostrare che **INDSET** appartiene a **NP**.
c) Definire il linguaggio **CLIQUE**.
d) Dimostrare che **CLIQUE** $\leq_p$ **INDSET**, fornendo una opportuna **funzione di riduzione**.
e) Cosa possiamo **dedurre** per INDSET dalle affermazioni b) e d)?

d)

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 32 pag. 29

$CLIQUE$ è NP-completo

e)

Teorema

$CLIQUE$ è NP-completo.

Dimostrazione.

- Sappiamo che $CLIQUE \in NP$.
- Inoltre, $3SAT$ è NP-completo e $3SAT \leq_P CLIQUE$
- Quindi $CLIQUE$ è NP-completo.

$\square$

Lezione 32 pag. 22

Provare la *NP* – completezza

e) Teorema

Se B è NP-completo e $B \leq_p C$, con $C \in NP$, allora C è NP-completo.

Una possibile strategia per provare che un linguaggio C è NP-completo:

1. Mostrare che $C \in NP$
2. Scegliere un linguaggio B che sia NP-completo
3. Definire una riduzione di tempo polinomiale di B in C.

Fine dei tutorati

(per questo A.A.)

buono studio…
e in bocca al lupo
per le vostre prossime prove ☺