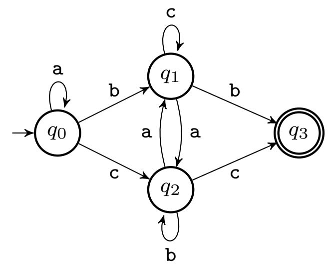
Automi e Linguaggi (M. Cesati)

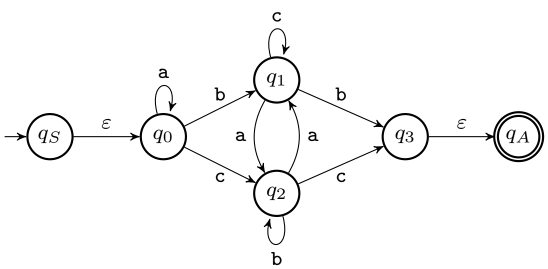
Facoltà di Ingegneria, Università degli Studi di Roma Tor Vergata

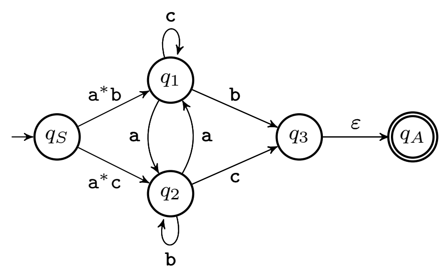
Compito scritto del 26 agosto 2022

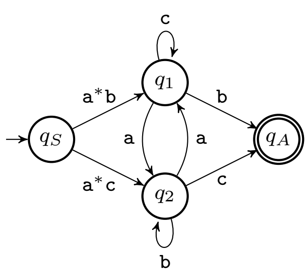
Esercizio 1 [6] Determinare una espressione regolare per il linguaggio riconosciuto dal DFA:

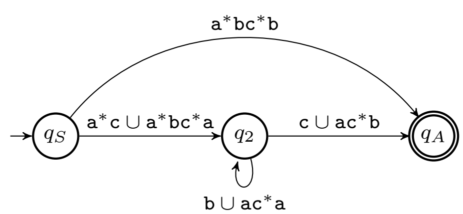


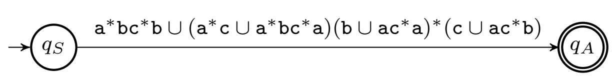
Soluzione: Convertiamo il DFA in un GNFA e rimuoviamo nell'ordine i nodi q_0, q_3, q_1 e q_2. Si ottiene:











In conclusione, una espressione regolare è $\mathtt{a}^*[\mathtt{bc^*b} \cup (\mathtt{c} \cup \mathtt{bc^*a})(\mathtt{b} \cup \mathtt{ac^*a})^*(\mathtt{c} \cup \mathtt{ac^*b})]$,

Esercizio 2 [6] Si consideri il linguaggio $A = \left\{ w\, z\, \overline{w}^{\mathcal{R}} \mid w \in \{0,1\}^*, z = 0^n 1^n, n \geq 0 \right\}$, ove $\overline{w}^{\mathcal{R}}$ è la stringa ottenuta complementando i bit in w ed invertendone l'ordine. Il linguaggio A è regolare? Giustificare la risposta con una dimostrazione.

Soluzione: Il linguaggio A non è regolare. Tuttavia questa dimostrazione non può essere basata sul fatto che A contiene, come sottoinsieme, un linguaggio non regolare (se bastasse questo, potremmo anche dimostrare che $\{0,1\}^*$ è non regolare!). Invece, assumiamo per assurdo che l'intero linguaggio A sia regolare, e dunque che per esso valga il pumping lemma con lunghezza p. Consideriamo la stringa $s = 0^p1^p$, che fa certamente parte di A considerando, ad esempio, $w = \varepsilon$ e $n = p$. Il pumping lemma afferma che esiste una suddivisione $s = xyz$ con $|xy| \leq p$, $|y| > 0$ e $xy^iz \in A$ per ogni $i \geq 0$. Consideriamo quindi il caso $i = 2$, e quindi la stringa $xyyz$. Poiché $|xy| \leq p$, y contiene solo zeri, quindi $xyyz = 0^q1^p$ con $q > p$. Tuttavia, il linguaggio A non contiene alcuna stringa della forma 0^q1^p con $q > p$.

Infatti, supponiamo per assurdo che $0^q1^p \in A$. Quindi $0^q1^p = wz\overline{w}^{\mathcal{R}}$. Se $w = \varepsilon$, allora $0^q1^p = z = 0^n1^n$, per qualche n, quindi varrebbe $q = p$, contraddicendo che $q > p$. Se invece $w \neq \varepsilon$, allora

$$0^q1^p = \underbrace{0^l1^m}_{w}\underbrace{0^n1^n}_{z}\underbrace{0^m1^l}_{\overline{w}^{\mathcal{R}}}$$

con $m+l > 0$ e $n \geq 0$. Quindi necessariamente, per evitare l'occorrenza di un 1 seguito da uno 0, $l > 0$ e $m = 0$. Perciò $0^q1^p = 0^{l+n}1^{l+n}$, e quindi $q = p$, contraddicendo la condizione $q > p$. La contraddizione deriva dall'aver supposto che $0^q1^p \in A$ con $q > p$. Un altro ragionamento che porta alla stessa conclusione si basa sull'osservazione che il numero di bit 0 ed il numero di bit 1 in ogni stringa appartenente ad A deve coincidere: infatti per definizione nella porzione z della stringa vi sono lo stesso numero di 0 e 1, e lo stesso vale nella stringa $w\overline{w}^{\mathcal{R}}$, in cui ad ogni bit in w corrisponde un bit complementato in $\overline{w}^{\mathcal{R}}$. Pertanto, $0^q1^p \notin A$ se $q \neq p$.

Poiché la stringa pompata non può appartenere ad A, il pumping lemma non vale, e dunque A non può essere regolare.

Esercizio 3 [6] Si consideri la grammatica G con variabile iniziale S:

$$S \to A \quad A \to \mathtt{a}A\mathtt{b} \mid \mathtt{b}A\mathtt{a} \mid \mathtt{a}B\mathtt{b} \quad B \to \mathtt{a}B\mathtt{b} \mid \varepsilon.$$

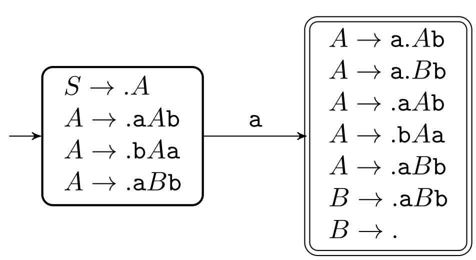
La grammatica G è deterministica? Giustificare la risposta con una dimostrazione.

Soluzione: La grammatica G non può essere deterministica in quanto essa è ambigua. È sufficiente infatti considerare le seguenti due differenti derivazioni "a sinistra" della stringa aabb:

$$S \Rightarrow A \Rightarrow \mathtt{a}A\mathtt{b} \Rightarrow \mathtt{aa}B\mathtt{bb} \Rightarrow \mathtt{aa}\varepsilon\mathtt{bb} = \mathtt{aabb}$$

$$S \Rightarrow A \Rightarrow \mathtt{a}B\mathtt{b} \Rightarrow \mathtt{aa}B\mathtt{bb} \Rightarrow \mathtt{aa}\varepsilon\mathtt{bb} = \mathtt{aabb}$$

Come dimostrazione alternativa è sufficiente esibire due stati dell'automa DK:



Poiché lo stato di accettazione contiene una regola in cui il punto è seguito da un simbolo terminale, G non è deterministica.

Esercizio 4 [6] Sia $B = \{ \langle M \rangle \mid M$ è una macchina di Turing tale che per ogni stringa x accettata da M, M accetta anche la stringa rovesciata $x^{\mathcal{R}} \}$. Il linguaggio B è decidibile? Giustificare la risposta con una dimostrazione.

Soluzione: Poiché il linguaggio è costituito da codifiche di macchine di Turing, è plausibile che ad esso possa applicarsi il teorema di Rice, che afferma che qualunque proprietà non banale relativa ai linguaggi riconosciuti da TM è indecidibile. Per verificare se le ipotesi del teorema di Rice sono soddisfatte, consideriamo se la proprietà caratterizzante l'insieme B è banale o meno, ovvero se B è diverso dall'insieme vuoto e dall'insieme di tutte le codifiche delle TM. Innanzi tutto, sia M' una TM che accetta esclusivamente la stringa 00 e rifiuta tutte le altre stringhe; poiché $(00)^{\mathcal{R}} = 00$, $\langle M' \rangle \in B$, dunque $B \neq \emptyset$. D'altra parte, sia M'' una TM che accetta esclusivamente la stringa 01 e rifiuta tutte le altre stringhe: ovviamente $(01)^{\mathcal{R}} = 10 \notin L(M'')$, dunque $\langle M'' \rangle \notin B$.

La seconda ipotesi del teorema di Rice è che la proprietà caratterizzante B deve riferirsi al linguaggio riconosciuto dalle macchine di Turing, e non alle TM stesse. Ciò è evidente dalla definizione stessa di B: se $\langle M \rangle \in B$ e $L(N) = L(M)$, allora necessariamente N deve accettare il rovescio di qualunque stringa $x \in L(N)$ perché $x^{\mathcal{R}} \in L(M) = L(N)$; dunque $\langle N \rangle \in B$.

Avendo quindi verificato le ipotesi del teorema di Rice, possiamo concludere direttamente con il suo asserto, ossia che il linguaggio B non è decidibile.

È possibile anche dimostrare che B è non decidibile tramite una riduzione diretta da un altro problema non decidibile, quale ad esempio $\mathcal{A}_{\mathrm{TM}}$ ($\mathcal{A}_{\mathrm{TM}} \leq_m B$). Si deve prestare attenzione però a non confondere le istanze dei due problemi. Ad esempio, una TM T che rifiuta ogni input (ossia tale che $L(T) = \emptyset$) in effetti appartiene a B, perché se T accetta una stringa accetta anche il suo rovescio (banalmente è vero perché T non accetta alcuna stringa); quindi $\langle T \rangle \in B$. D'altra parte, $\langle T, x \rangle \notin \mathcal{A}_{\mathrm{TM}}$ per ogni possibile input x.

Supponiamo quindi che esista, per assurdo, un decisore D per il linguaggio B, e consideriamo la seguente TM:

P= "On input $\langle M, w \rangle$, where M is a TM and w is a string:
1. From $\langle M, w \rangle$, build the encoding of the following DTM R:

    R="On any input x:
    a. If $x = 01$, then accepts
    b. Run $M(w)$
    d. If $M(w)$ accepts, then accept
    e. If $M(w)$ rejects, then reject"

2. Run D on input $\langle R \rangle$
3. If $D(\langle R \rangle)$ accepts, then accept, else reject"

Supponiamo che $M(w)$ accetti; dunque $R(x)$ accetta sia se $x = 01$ sia se $x \neq 01$; perciò R accetta ogni stringa, ossia $L(R) = \Sigma^*$. Di conseguenza, $\langle R \rangle \in B$. Se invece $M(w)$ non accetta (sia perché rifiuta oppure perché non si ferma), allora $R(x)$ accetta soltanto se $x = 01$; dunque $L(R) = \{01\}$, e $\langle R \rangle \notin B$, in quanto $10 \notin L(R)$. Il decisore D può determinare se $\langle R \rangle \in B$, e di conseguenza P può decidere la generica istanza $\langle M, w \rangle$ di $\mathcal{A}_{\mathrm{TM}}$. Questa è ovviamente una contraddizione perché $\mathcal{A}_{\mathrm{TM}}$ non è decidibile.

Esercizio 5 [7] Siano A e B linguaggi Turing-riconoscibili (ossia ricorsivamente enumerabili). La differenza simmetrica $A \triangle B$ di A e B (gli elementi che stanno in A o in B ma non in entrambi) è necessariamente Turing-riconoscibile? Giustificare la risposta con una dimostrazione.

Soluzione: $A \triangle B$ non è necessariamente Turing-riconoscibile; per dimostrarlo è sufficiente esibire un contro-esempio. Sia dunque $A = \mathcal{A}_{\mathrm{TM}}$, il linguaggio contenente le codifiche delle macchine di Turing e delle stringhe da esse accettate. Sia inoltre $B = \Sigma^*$, ove Σ è l'alfabeto sul quale sono costruite le codifiche delle TM in $\mathcal{A}_{\mathrm{TM}}$. Si dimostra facilmente che $A \triangle B = (A \setminus B) \cup (B \setminus A)$; inoltre nel nostro caso $\mathcal{A}_{\mathrm{TM}} \setminus \Sigma^* = \emptyset$ e $\Sigma^* \setminus \mathcal{A}_{\mathrm{TM}} = {\mathcal{A}_{\mathrm{TM}}}^c$. Perciò $\mathcal{A}_{\mathrm{TM}} \triangle \Sigma^* = \emptyset \cup {\mathcal{A}_{\mathrm{TM}}}^c = {\mathcal{A}_{\mathrm{TM}}}^c$. Sappiamo che Σ^* è regolare e quindi Turing-riconoscibile; anche $\mathcal{A}_{\mathrm{TM}}$ è Turing-riconoscibile, ma non decidibile. Perciò $\Sigma^* \triangle \mathcal{A}_{\mathrm{TM}} = {\mathcal{A}_{\mathrm{TM}}}^c$ non può essere Turing-riconoscibile; se infatti lo fosse, poiché sia $\mathcal{A}_{\mathrm{TM}}$ che ${\mathcal{A}_{\mathrm{TM}}}^c$ sarebbero Turing-riconoscibili, allora sarebbero anche entrambi decidibili, il che è manifestamente falso.

Esercizio 6 [9] Si consideri un problema in cui l'istanza è costituita da $2\,m$ numeri interi non negativi (non necessariamente distinti tra loro) la cui somma è pari ($2\,s$). Il problema richiede di decidere se è possibile suddividere i numeri in due sottoinsiemi ciascuno con m elementi e tali che la somma degli elementi in ciascun sottoinsieme sia uguale a s. Dimostrare che tale problema è NP-completo.

Soluzione: Questo problema è conosciuto col nome di BALANCED PARTITION. È un problema polinomialmente verificabile: infatti, sia U un multi-insieme di numeri interi non negativi con somma $2\,s$ che ammette una partizione in due multi-insiemi I e J ($I \cup J = U$, $I \cap J = \emptyset$) tale che $\sum_{x \in I} x = \sum_{x \in J} x = s$. Un certificato per tale istanza-sì di BALANCED PARTITION è costituito semplicemente da uno dei due sottoinsiemi. Esiste dunque un verificatore che opera in tempo polinomiale nella dimensione dell'istanza:

V= "On input $\langle U, I \rangle$, where U is a multi-set of non-negative integers:
1. Verify that $I \subset U$ and $2\,|I| = |U|$

2. Compute $J = U \setminus I$
3. Compute $v = \sum_{x \in I} x$
4. Compute $w = \sum_{x \in J} x$
5. Accept if $v = w$, reject otherwise"

Pertanto, Balanced Partition è incluso in NP. Per dimostrare che è anche NP-hard, possiamo esibire una riduzione polinomiale da un altro problema NP-hard, ad esempio Subset Sum.

Sia dunque (S, t) una istanza di Subset Sum, ossia un multi-insieme di numeri interi S ed un intero t. Dalla dimostrazione di NP-hardness fatta a lezione è evidente che questo problema è NP-hard anche restrigendo le istanze agli interi non negativi. Consideriamo la riduzione polinomiale che trasforma (S, t) nel multi-insieme U come segue. Sia $n = |S|$ e sia $\mu = \sum_{x \in S} x$. Se $\mu < t$, allora (S, t) è certamente una istanza-no, dunque la riduzione polinomiale si limita a costruire una istanza-no elementare di Balanced Partition. Altimenti, il multi-insieme U è costituito da S, dall'intero non negativo $\lambda = 2\,t - \mu$, e da $n + 1$ valori interi nulli (ossia n zeri $z_0 = \cdots = z_n = 0$). Ovviamente $|U| = |S| + 1 + n + 1 = 2\,(n + 1)$.

Supponiamo che (S, t) sia una istanza-sì di Subset Sum, e dunque che esista $T \subseteq S$ tale che $\sum_{x \in T} x = t$. Sia $q = |T|$, e consideriamo il multi-insieme $W = T \cup \{z_0, \ldots z_{n-q}\}$ (si osservi che se $q = n$ allora W include il solo elemento z_0). Quindi $|W| = n+1$. Inoltre il sottoinsieme $Z = U \setminus W$ è costituito da $n-q$ elementi di $S \setminus T$, da λ, e da q zeri $\{z_{n-q+1}, \ldots z_n\}$ (ovviamente $|Z| = (n - q) + 1 + (n - n + q - 1 + 1) = n + 1$). Si ha:

$$\sum_{x \in W} x = \sum_{x \in T} x + z_0 + \cdots z_{n-q} = t$$

e

$$\sum_{x \in Z} x = \sum_{x \in S \setminus T} x + \lambda + z_{n-q+1} + \cdots + z_n = (\mu - t) + (2\,t - \mu) = t.$$

Pertanto U è una istanza-sì di Balanced Partition.

Supponiamo al contrario che U sia una istanza-sì di Balanced Partition derivata dalla riduzione di una istanza (S, t), e siano W e Z tali che $W \cap Z = \emptyset$, $|W| = |Z| = n + 1$, e $\sum_{x \in W} x = \sum_{x \in Z} x = t$. Senza perdita di generalità supponiamo che $\lambda \notin W$, e sia $T = W \setminus \{z_0, \ldots, z_n\}$, ossia T è il multi-insieme W a cui sono stati rimossi gli zeri z_i eventualmente presenti. Pertanto $T \subseteq S$, ed inoltre $\sum_{x \in T} x = t$. Perciò (S, t) è una istanza-sì di Subset Sum.