

1. (12 punti) Data una stringa w di 0 e 1, il *flip* di w si ottiene cambiando tutti gli 0 in w con 1 e tutti gli 1 in w con 0. Dato un linguaggio L , il flip di L è il linguaggio

$$\text{flip}(L) = \{w \in \{0, 1\}^* \mid \text{il flip di } w \text{ appartiene ad } L\}.$$

Dimostra che la classe dei linguaggi regolari è chiusa rispetto all'operazione di flip.

Soluzione: Se L è un linguaggio regolare, allora sappiamo che esiste un DFA $A = (Q, \Sigma, \delta, q_0, F)$ che riconosce L . Costruiamo un DFA $A' = (Q', \Sigma, \delta', q'_0, F')$ che accetta il linguaggio $\text{flip}(L)$ come segue.

- $Q' = Q$. L'insieme degli stati rimane lo stesso.
- L'alfabeto Σ rimane lo stesso.
- Per ogni stato $q \in Q$, $\delta'(q, 0) = \delta(q, 1)$ e $\delta'(q, 1) = \delta(q, 0)$. La funzione di transizione scambia gli 0 con 1 e gli 1 con 0.
- $q'_0 = q_0$. Lo stato iniziale non cambia.
- $F' = F$. Gli stati finali rimangono invariati.

Per dimostrare che A' riconosce il linguaggio $\text{flip}(L)$, dobbiamo considerare due casi.

- Se $w \in \text{flip}(L)$, allora sappiamo che il flip di w appartiene ad L . Chiamiamo \bar{w} il flip di w . Siccome A riconosce L , allora esiste una computazione di A che accetta \bar{w} :

$$s_0 \xrightarrow{\bar{w}_1} s_1 \xrightarrow{\bar{w}_2} \dots \xrightarrow{\bar{w}_n} s_n$$

con $s_0 = q_0$ e $s_n \in F$. Se scambiamo gli zeri e gli uni nella computazione, otteniamo una computazione accettante per A' sulla parola w . Di conseguenza, abbiamo dimostrato che $w \in L(A')$.

- Viceversa, se w è accettata dal nuovo automa A' , allora esiste una computazione accettante che ha la forma

$$s_0 \xrightarrow{w_1} s_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} s_n$$

con $s_0 = q_0$, $s_n \in F'$. Se scambiamo gli zeri e gli uni nella computazione, otteniamo una computazione accettante per A sul flip di w . Di conseguenza, il flip di w appartiene ad L e abbiamo dimostrato che $w \in \text{flip}(L)$.

2. (12 punti) Considera il linguaggio

$$L_2 = \{w \in \{0,1\}^* \mid w \text{ non è palindroma}\}.$$

Una parola è *palindroma* se rimane uguale letta da sinistra a destra e da destra a sinistra. Dimostra che L_2 non è regolare.

Soluzione: Consideriamo il complementare del linguaggio L_2 , ossia il linguaggio

$$\overline{L_2} = \{w \in \{0,1\}^* \mid w \text{ è palindroma}\}.$$

Usiamo il Pumping Lemma per dimostrare che il linguaggio $\overline{L_2}$ non è regolare. Supponiamo per assurdo che lo sia:

- sia k la lunghezza data dal Pumping Lemma;
- consideriamo la parola $w = 0^k 10^k$, che è di lunghezza maggiore di k ed è palindroma;
- sia $w = xyz$ una suddivisione di w tale che $y \neq \varepsilon$ e $|xy| \leq k$;
- poiché $|xy| \leq k$, allora x e y sono entrambe contenute nella sequenza iniziale di 0. Inoltre, siccome $y \neq \varepsilon$, abbiamo che $x = 0^q$ e $y = 0^p$ per qualche $q \geq 0$ e $p > 0$. z contiene la parte rimanente della stringa: $z = 0^{k-q-p} 10^k$. Consideriamo l'esponente $i = 2$: la parola xy^2z ha la forma

$$xy^2z = 0^q 0^{2p} 0^{k-q-p} 10^k = 0^{k+p} 10^k$$

La parola iterata xy^2z non appartiene ad $\overline{L_2}$ perché non è palindroma: se la rovesciamo diventa la parola $0^k 10^{k+p}$ che è una parola diversa perché $p > 0$.

Abbiamo trovato un assurdo quindi $\overline{L_2}$ non può essere regolare.

Siccome i linguaggi regolari sono chiusi per complementazione, nemmeno L_2 può essere regolare.

3. (12 punti) Dimostra che se $L \subseteq \Sigma^*$ è un linguaggio context-free allora anche il seguente linguaggio è context-free:

$$\text{dehash}(L) = \{\text{dehash}(w) \mid w \in L\},$$

dove $\text{dehash}(w)$ è la stringa che si ottiene cancellando ogni $\#$ da w .

Soluzione: Se L è un linguaggio context-free, allora esiste una grammatica $G = (V, \Sigma, R, S)$ che lo genera. Possiamo assumere che questa grammatica sia in forma normale di Chomsky. Per dimostrare che $\text{dehash}(L)$ è context-free, dobbiamo essere in grado di definire una grammatica che possa generarlo. Questa grammatica è una quadrupla $G' = (V', \Sigma', R', S')$ definita come segue.

- L'alfabeto tutti i simboli di Σ tranne $\#$: $\Sigma' = \Sigma \setminus \{\#\}$.
- L'insieme di variabili è lo stesso della grammatica G : $V' = V$.

- Il nuovo insieme di regole R' è ottenuto rimpiazzando ogni regola nella forma $A \rightarrow \#$ con la regola $A \rightarrow \varepsilon$, e lasciando invariate le regole nella forma $A \rightarrow BC$, le regole nella forma $A \rightarrow b$ quando $b \neq \#$, e la regola $S \rightarrow \varepsilon$ (se presente).
- La variabile iniziale rimane la stessa: $S' = S$.

Data una derivazione $S \Rightarrow^* w$ della grammatica G possiamo costruire una derivazione nella nuova grammatica G' che applica le stesse regole nello stesso ordine, e che deriva una parola dove ogni $\#$ è rimpiazzato dalla parola vuota ε . Quindi, G' permette di derivare tutte le parole in $dehash(L)$.

Viceversa, data una derivazione $S \Rightarrow^* w$ della nuova grammatica G' possiamo costruire una derivazione nella grammatica G che applica le stesse regole nello stesso ordine. Di conseguenza, in ogni punto in cui la derivazione per G' applica la regola modificata $A \rightarrow \varepsilon$, la derivazione per G applicherà la regola $A \rightarrow \#$ inserendo un $\#$ in qualche punto della parola w . Al termine della derivazione si ottiene una parola w' tale che $dehash(w') = w$. Quindi, G' permette di derivare solo parole che appartengono a $dehash(L)$.