Automi e Linguaggi Formali – 26/6/2023
Primo appello – Prima Parte: Linguaggi regolari e linguaggi contex-free

1. (12 punti) Diciamo che una stringa x è un *prefisso* della stringa y se esiste una stringa z tale che $xz = y$, e che è un *prefisso proprio* di y se vale anche $x \neq y$. Dimostra che se $L \subseteq \Sigma^*$ è un linguaggio regolare allora anche il linguaggio

$$NOPREFIX(L) = \{w \in L \mid \text{ nessun prefisso proprio di } w \text{ appartiene ad } L\}$$

è un linguaggio regolare.

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' che accetta il linguaggio $NOPREFIX(L)$, aggiungendo uno stato "pozzo" agli stati di A, ossia uno stato non finale q_s tale che la funzione di transizione obbliga l'automa a rimanere per sempre in q_s una volta che lo si raggiunge. Lo stato iniziale e gli stati finali rimangono invariati. La funzione di transizione di A' si comporta come quella di A per gli stati non finali, mentre va verso lo stato pozzo per qualsiasi simbolo dell'alfabeto a partire dagli stati finali. In questo modo le computazioni accettanti di A' sono sempre sequenze di stati dove solo l'ultimo stato è finale, mentre tutti quelli intermedi sono non finali. Di conseguenza le parole che A' accetta sono accettate anche da A, mentre tutti i prefissi propri sono parole rifiutate da A, come richiesto dalla definizione del linguaggio $NOPREFIX(L)$.

Formalmente, $A' = (Q', \Sigma, \delta', q_0', F')$ è definito come segue.

- $Q' = Q \cup \{q_s\}$, con $q_s \notin Q$.
- L'alfabeto Σ rimane lo stesso.
- $\delta'(q, a) = \begin{cases} \delta(q, a) & \text{se } q \notin F \\ q_s & \text{altrimenti} \end{cases}$
- $q_0' = q_0$. Lo stato iniziale non cambia.
- $F' = F$. Gli stati finali rimangono invariati.

Per dimostrare che A' riconosce il linguaggio $NOPREFIX(L)$, dobbiamo considerare due casi.

- Se $w \in NOPREFIX(L)$, allora sappiamo che $w \in L$, mentre nessun prefisso proprio di w appartiene ad L. Di conseguenza esiste una computazione di A che accetta la parola:
$$s_0 \xrightarrow{w_1} s_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} s_n$$
con $s_0 = q_0$ e $s_n \in F$. Siccome tutti i prefissi propri di w sono rifiutati da A, allora gli stati $s_0, \dots, s_{n-1}$ sono tutti non finali. Per la definizione di A', la computazione che abbiamo considerato è anche una computazione accettante per A', e di conseguenza, $w \in L(A')$.
- 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 $s_0 = q_0$, $s_n \in F$ e dove tutti gli stati intermedi $s_0, \ldots, s_{n-1}$ sono non finali. Di conseguenza, la computazione è una computazione accettante anche per A, quindi $w \in L$. Siccome tutti gli stati intermedi della computazione sono non finali, allora A rifiuta tutti i prefissi propri di w, e quindi $w \in NOPREFIX(L)$.

2. **(12 punti)** Considera il linguaggio

$$L_2 = \{uvvu \mid u, v \in \{0,1\}^*\}.$$

Dimostra che L_2 non è regolare.

Soluzione: Usiamo il Pumping Lemma per dimostrare che il linguaggio non è regolare. Supponiamo per assurdo che L_2 sia regolare:

- sia k la lunghezza data dal Pumping Lemma;
- consideriamo la parola $w = 0^k 110^k$, che è di lunghezza maggiore di k ed appartiene ad L_2 perché la possiamo scrivere come $uvvu$ ponendo $u = 0^k$ e $v = 1$;
- 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: $z = 0^{k-q-p}110^k$. Consideriamo l'esponente $i = 2$: la parola xy^2z ha la forma

$$xy^2z = 0^q 0^{2p} 0^{k-q-p} 110^k = 0^{k+p} 110^k$$

La parola iterata xy^2z non appartiene ad L_2 perché non si può scrivere nella forma $uvvu$. Visto che la parte iniziale deve essere uguale a quella finale, si deve porre $u = 0^k$, ma in questo caso la parte centrale della parola è 0^p11 che non si può dividere in due metà uguali. Viceversa, se si pone $v = 1$ per avere la parte centrale della parola composta da due metà uguali, allora si ottiene una sequenza iniziale di 0 che è più lunga della sequenza finale di 0.

Abbiamo trovato un assurdo quindi L_2 non può essere regolare.

3. **(12 punti)** Una grammatica context-free è *lineare* se ogni regola in R è nella forma $A \to aBc$ o $A \to a$ per qualche $a, c \in \Sigma \cup \{\varepsilon\}$ e $A, B \in V$. I linguaggi generati dalle grammatiche lineari sono detti *linguaggi lineari*. Dimostra che i linguaggi regolari sono un sottoinsieme proprio dei linguaggi lineari.

Soluzione. Per risolvere l'esercizio dobbiamo dimostrare che ogni linguaggio regolare è anche un linguaggio lineare (i linguaggi regolari sono un sottoinsieme dei linguaggi lineari), e che esistono linguaggi lineari che non sono regolari (l'inclusione è propria).

- Dato un linguaggio regolare L, sappiamo che esiste un DFA $A = (Q, \Sigma, \delta, q_0, F)$ che riconosce L. Inoltre, sappiamo che ogni DFA può essere convertito in una grammatica context-free dove le regole sono del tipo $R_i \to aR_j$ per ogni transizione $\delta(q_i, a) = q_j$ del DFA, e del tipo $R_i \to \varepsilon$ per ogni stato finale del q_i del DFA. Entrambi i tipi di regola rispettano le condizioni di linearità, quindi la grammatica equivalente al DFA è lineare, e questo implica che L è un linguaggio lineare.

- Consideriamo il linguaggio non regolare $L = \{0^n 1^n \mid n \geq 0\}$. La seguente grammatica lineare genera L:

  $$S \to 0S1 \mid \varepsilon$$

  Quindi, esiste un linguaggio lineare che non è regolare.

BASTA USARE UN LINGUAGGIO NON REGOLARE !