Automi e Linguaggi Formali – 20/4/2022
Prima prova intermedia – Secondo Turno – Soluzione

1. (12 punti) Se L è un linguaggio e a un simbolo, allora L/a, il *quoziente* di L e a, è l'insieme delle stringhe

$$L/a = \{w \mid wa \in L\}.$$

Per esempio, se $L = \{a, aab, baa\}$, allora $L/a = \{\varepsilon, ba\}$. Dimostra che se L è regolare allora anche L/a è regolare.

Soluzione: Se L è un linguaggio regolare, allora sappiamo che esiste un DFA $A = (Q, \Sigma, \delta, q_0, F)$ che riconosce L. Data una parola $wa \in L$ dove $w = w_1 \dots w_n$, la computazione di A su aw è una sequenza di stati $r_0 r_1 \dots r_{n+1}$ tali che:

- $r_0 = q_0$;
- $\delta(r_{i-1}, w_i) = r_i$ per ogni $i = 1, \dots, n$;
- $\delta(r_n, a) = r_{n+1}$;
- $r_{n+1} \in F$.

Data questa osservazione possiamo costruire un automa A' che accetta il linguaggio L/a cambiando gli stati finali di A. Formalmente, $A' = (Q, \Sigma, \delta, q_0, F')$ dove Q, Σ, δ e q_0 sono gli stessi di A e l'insieme degli stati finali contiene tutti gli stati che raggiungono uno stato finale di A dopo aver consumato a:

$$F' = \{q \mid \delta(q, a) \in F\}.$$

In questo modo abbiamo che per ogni $wa \in L$ la sequenza di stati $r_0 \dots r_n$ descritta sopra è una computazione di A' che accetta w (perché r_n diventa uno stato finale di A'), ed abbiamo dimostrato che se $wa \in L$ allora $w \in L(A')$. Viceversa, se $w \in L(A')$ allora esiste una computazione $s_0 \dots s_n$ di A' tale che:

- $s_0 = q_0$;
- $\delta(s_{i-1}, w_i) = s_i$ per ogni $i = 1, \dots, n$;
- $s_n \in F'$.

Di conseguenza, $s_{n+1} = \delta(s_n, a) \in F$ e la sequenza di stati $s_1 \dots s_{n+1}$ è una computazione di A su wa, ed abbiamo dimostrato che se $w \in L(A')$ allora $wa \in L$. Quindi possiamo concludere che il linguaggio di A' è precisamente L/a, come richiesto.

2. (12 punti) Se w è una stringa di 0 e 1, allora $\overline{w}$ è una stringa formata da w sostituendo gli 0 con 1 e viceversa; per esempio $\overline{011} = 100$. Considera il linguaggio

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

Dimostra che L_2 non è regolare.

Soluzione: Prima di procedere con la soluzione ci è utile osservare che data una qualsiasi parola w, la parola $\overline{w}$ avrà sempre un numero di 0 uguale al numero di 1 di w, ed un numero di 1 uguale al numero di 0 di w. Di conseguenza, ogni parola nella forma $w\overline{w}$ avrà un numero di 0 uguale al numero di 1.

Ora possiamo usare 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 $w = 0^k 1^k$, che appartiene ad L_2 perché $\overline{0^k} = 1^k$, ed è di lunghezza maggiore di k;
- 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 $y \neq \emptyset$, abbiamo che $x = 0^q$ e $y = 0^p$ per qualche $q \geq 0$ e $p > 0$. z contiene la parte rimanente della stringa: $z = 0^{k-q-p} 1^k$. Consideriamo l'esponente $i = 2$: la parola xy^2z ha la forma

$$xy^2z = 0^q 0^{2p} 0^{k-q-p} 1^k = 0^{k+p} 1^k$$

  Poiché $p > 0$, la parola iterata xy^2z contiene più 0 che 1 e di conseguenza non può essere scritta nella forma $w\overline{w}$.

Abbiamo trovato un assurdo quindi L_2 non può essere regolare.

Nota: per questo esercizio scegliere qualsiasi esponente $i \neq 1$ permette di arrivare all'assurdo.

3. (12 punti) Dimostra che il linguaggio L_2 dell'esercizio precedente non è nemmeno un linguaggio context-free.

Soluzione: Possiamo usare il Pumping Lemma per linguaggi Context-Free per dimostrare che il linguaggio non è context-free. Supponiamo per assurdo che lo sia.

- Sia k la lunghezza data dal Pumping Lemma.
- Questa volta scegliere la parola è meno ovvio. Osserviamo per prima cosa che la parola $0^k 1^k$ si può iterare, dividendola come segue, e perciò non è adatta al nostro scopo:

$$\overbrace{\underbrace{000\ldots000}_{u}\underbrace{0}_{v}}^{0^k}\underbrace{\varepsilon}_{x}\overbrace{\underbrace{1}_{y}\underbrace{111\ldots111}_{z}}^{1^k}$$

  Anche la parola $0^k 1^k 1^k 0^k$ si può iterare, suddividendola in modo simile:

$$\overbrace{\underbrace{000\ldots000}_{u}\underbrace{0}_{v}}^{0^k}\underbrace{\varepsilon}_{x}\overbrace{\underbrace{1}_{y}\underbrace{111\ldots111000\ldots000}_{z}}^{1^k 1^k 0^k}$$

  Mostriamo invece che la parola $w = 0^k 10^k 1^k 01^k$ non può essere iterata;
- sia $w = uvxyz$ una suddivisione di w tale che $|vy| > 0$ e $|vxy| \leq k$;
- mostriamo che la sottostringa vxy deve stare a cavallo del punto centrale di w. Altrimenti, se vxy è inclusa nella prima metà della stringa, la stringa uv^2xy^2z sposta uno 0 nella prima posizione della seconda metà e quindi essa non può essere nella forma $w\overline{w}$. Viceversa, se vxy è inclusa nella seconda metà di w, la stringa uv^2xy^2z sposta un 1 nell'ultima posizione della prima metà e quindi essa non può essere nella forma $w\overline{w}$.

  Ma se la sottostringa vxy è a cavallo del punto centrale di w, la stringa $uv^0xy^0z = uxz$ ha la forma $0^k 10^i 1^j 01^k$ dove i e j non possono essere entrambi k, e quindi non può essere nella forma $w\overline{w}$. Quindi w non può essere iterata.

Abbiamo trovato un assurdo quindi L_2 non può essere context-free.