Indice

1 Corrispondenza tra grammatiche di tipo 3 e automi a stati finiti

Teorema 1. *Dato un automa a stati finiti posso costruire una grammatica equivalente e viceversa. Per ora esclundendo le ϵ-regole e mosse.*

Dimostrazione. Iniziamo dal primo lato, da un automa

$$\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$$

costruiamo la grammatica che genera lo stesso linguaggio (salvo la parola vuota se questa è in $L(\mathcal{A})$). La grammatica è della forma

$$G = \langle Q, \Sigma, P, q_0 \rangle$$

Le produzioni sono costruite a partire dalle transizioni dell'automa: per ogni transizione $\delta(q, a) = p$, aggiungo la produzione $q \to ap$ e la produzione $q \to a$ se $p \in F$.

Ora mostriamo il caso opposto. Sia

$$G = \langle V, \Sigma, P, S \rangle$$

e l'automa corrispondente

$$\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$$

con

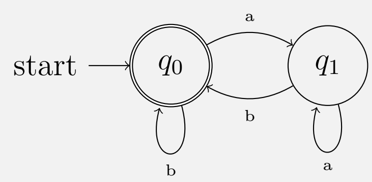
$$\begin{aligned} Q &= V \cup \{q_F\} \\ F &= \{q_F\} \\ q_0 &= S \end{aligned}$$

con q_F nuovo. Per ogni produzione in G definisco δ

$$\delta(A, a) = \begin{cases} B \in \delta(A, a) & \text{per } A \to aB \\ q_F \in \delta(A, a) & \text{per } A \to a \end{cases}$$

Ovviamente l'automa generato è non deterministico. ∎

Ad esempio costruiamo la grammatica corrispondente a



Questo corrisponde alla grammatica

$$G = \langle \{q_0, q_1\}, \{a, b\}, P, q_0 \rangle$$

con P definita come

$$\begin{aligned} q_0 &\to aq_1 \\ q_0 &\to bq_0 \\ q_1 &\to aq_1 \\ q_1 &\to bq_0 \\ q_1 &\to b \\ q_0 &\to b \end{aligned}$$

E si può dimostrare che

$$q_0 \overset{*}{\Rightarrow} xp, x \in \Sigma^* \Leftrightarrow \delta(q_0, x) = p$$

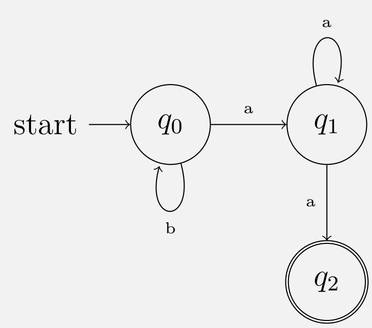
Data la grammatica

$$G = \langle \{S, A\}, \{a, b\}, P, S \rangle$$

con P definita come

$$\begin{aligned} S &\to aA \mid bS \\ A &\to aA \mid a \end{aligned}$$

Otteniamo



2 Limiti inferiori di automi non deterministici

Dimostriamo la non-regolarità di un linguaggio dal suo insieme di parole distinguibili

$$L = \{w \in \Sigma^* \mid \#_a(w) = \#_b(w)\}$$

intuitivamente visto che gli automi a stati finiti hanno una memoria finita, non possono contare un numero arbitrario di valori.
Si può dimostrare che per l'insieme

$$X = \{a^n \mid n \geq 0\}$$

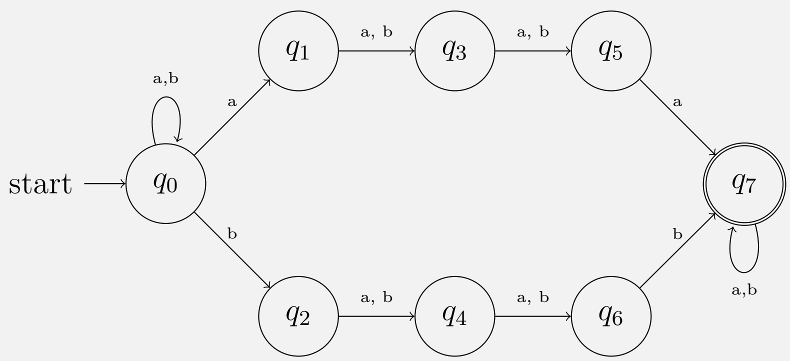
tutti i suoi elementi sono distinguibili, quindi L non può essere regolare.

Questo limite sul minimo numero di stati non vale per automi non deterministici.

Dato il linguaggio

$$L_n = \{x \in \{a, b\}^* \mid P(x)\}$$

Dove P è la proprietà per cui x contiene due simboli uguali tra loro a distanza n.
Ad esempio $ababba \in L_3$, infatti la seconda e l'ultima a e la prima e l'ultima b corrispondono. Costruiamo l'automa per L_3



Questo ha $2n+2$ stati, l'automa deterministico corrispondente deve ricordare la finestra degli utlimi n simboli, quindi dovrebbe avere almeno 2^n stati. Si può anche ragionare sull'insieme di stringhe distinguibili, infatti scegliamo

$$X = \{a, b\}^n$$

queste sono tutte distinguibili tra loro? Prendiamo due stringhe $x, y \in X$, con $x \neq y$, tali che

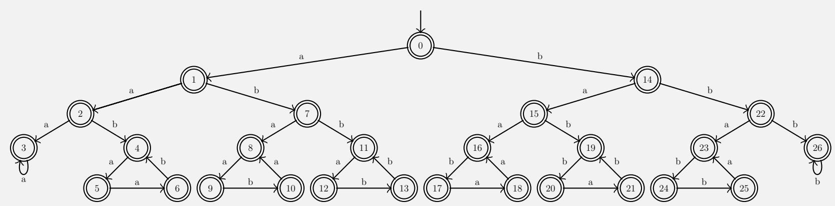
$$x = x_1 x_2 \dots x_i \dots x_n$$
$$y = y_1 y_2 \dots y_i \dots y_n$$

Sia i la prima la posizione in cui le due stringhe sono diverse. Sia $z = \overline{x_1}\ \overline{x_2} \dots \overline{x_{i-1}} a$. Quindi abbiamo trovato un insieme di 2^n stringhe distinguibili, quindi ogni DFA ha almeno 2^n stati.

Sia

$$L'_n = \{x \in \{a,b\}^* \mid P(x)\}$$

Dove P è la proprietà per cui per ogni coppia di simboli di x a distanza n i due coincidono.
Quindi per esempio $abbabba \in L'_3$. Alternativamente possiamo dire che $x \in L'_n$ sse $\exists w \in \{a,b\}^n$ e $\exists y$ prefisso di w per cui $\exists k \geq 0 \mid x = w^k y$.



Questo DFA è maggiore di 2^n stati.
Ma anche l'NFA corrispondente non ci permette di risparmiare un gran numero di spazi.
Come mai l'NFA corrispondente a questo e al precedente hanno una così grande differenza di stati? Il primo utilizza un esiste e questo un per ogni, e il non determinismo è utile per l'esiste.

Dato un linguaggio $L \subseteq \Sigma^*$ e $P \subseteq \Sigma^* \times \Sigma^* = \{(x_i, y_i) \mid i \in 1, \dots n\}$, diciamo che P è un **fooling set** per L sse valgono le seguenti condizioni

- $\forall i \in 1, \dots n \mid x_i y_i \in L$
- $\forall i, j \in 1, \dots n \mid i \neq j \Rightarrow x_i y_j \notin L$

Noi in particolare ci interessiamo all'*extended* fooling set, con la seconda condizione modificata:

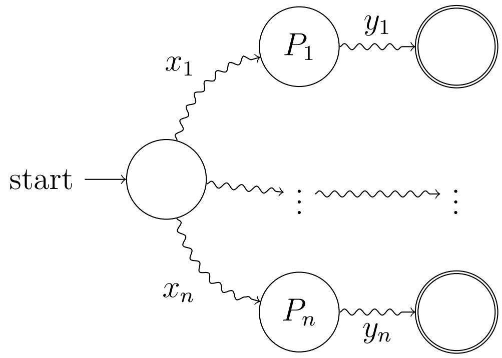
- $\forall i, j \in 1, \dots n \mid i \neq j \Rightarrow x_i y_j \notin L \vee x_j y_i \notin L$

Teorema 2. *Sia $L \subseteq \Sigma^*$ e P un extended fooling set per L, allora ogni automa non deterministico per L ha almeno $|P|$ stati.*

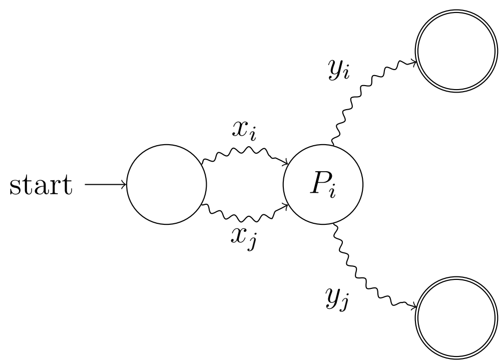
Dimostrazione. Sia $\mathcal{A}$ un NFA per L. Per ogni membro $(x_i, y_i) \in P$ abbiamo che

$$\delta(q_0, x_i) = P_i \text{ e } \delta(P_i, y_i) = F_i$$

con F_i finale.



Ora supponiamo per assurdo che per due elementi del fooling set P x_i e x_j portino entrambi nello stato P_i.



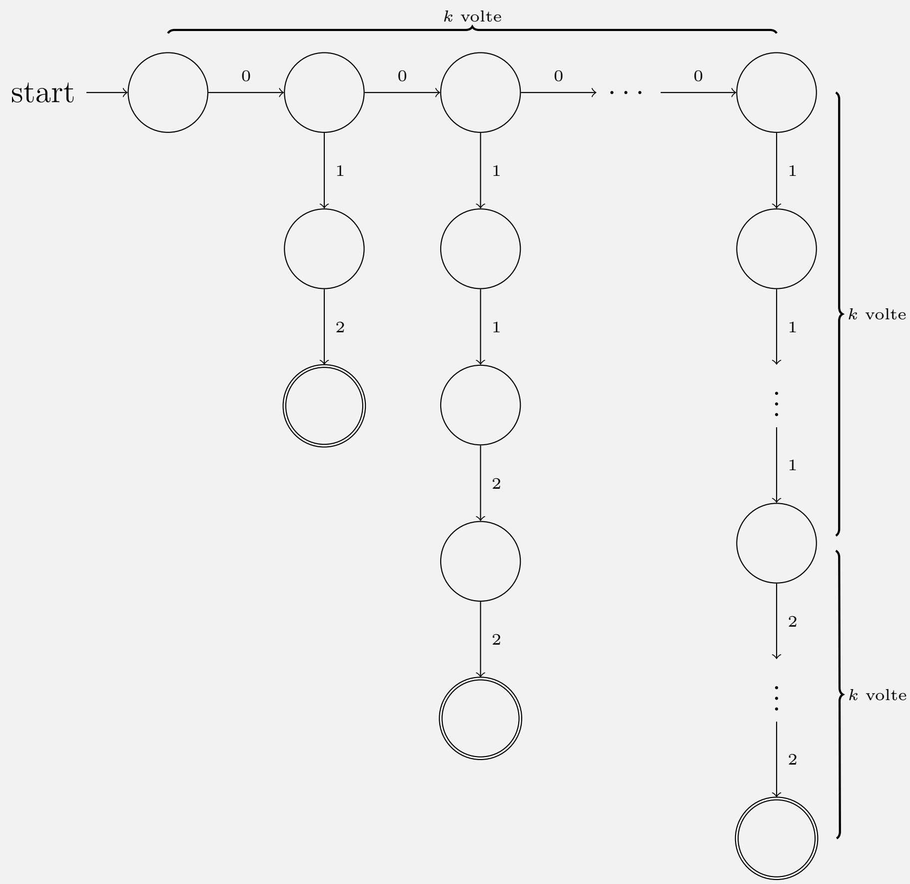
Allora è facile vedere che anche le parole x_iy_j e x_jy_i sarebbero accettare, infrangendo la seconda proprietà del fooling set. Quindi ad ogni x_i deve essere associato uno almeno uno stato che nessun'altro elemento condivide, quindi ci sono almeno $|P|$ stati. ■

Ritornando all'esempio di prima, sia $P = \{(x, x) \mid x \in \{a, b\}^n\}$, vale che ogni elemento $xx \in L'_n$, mentre per elementi diversi, $xy \notin L'_n$. Quindi questo è un fooling set valido per L'_n. E dato che l'NFA ha almeno $|P|$ stati, ne servono almeno 2^n.

Prendiamo il linguaggio

$$L_k = \{0^n 1^n 2^n \mid n \leq k\}$$

questo è rappresentabile dal DFA



Ma è facile mostrare che anche l'automa deterministico corrispondente non ci permette di risparmiare molto.