Indice

1 Numero di stati

Dato un linguaggio riconoscibile da un automa a stati finiti, quanti stati sono necessari per riconoscerlo?

Nell'ambito degli automi deterministici definiamo il concetto di **distribuibilità** tra stringhe rispetto al linguaggio $L \subseteq \Sigma^*$. Due stringhe qualunque $x, y \in \Sigma^*$ sono distinguibili per L se

$$\exists z \in \Sigma^* (xz \in L \land yz \notin L) \lor (xz \notin L \land yz \in L)$$

Molto semplicemente se x e y sono distinguibili queste ci porteranno in due stati diversi. Da questi due stati diversi si può costruire una z che in un caso ci porta in uno stato finale e nell'altro no. Quindi in un automa deterministico se due stringhe sono distinguibili, non ci possono mandare nello stesso stato.

Teorema 1. *Sia $L \subseteq \Sigma^*$ e $X \subseteq \Sigma^*$ tale che ogni coppia di stringhe in X è distinguibile in rispetto ad L. Allora ogni DFA che accetta L deve avere almeno $\#X$ stati (cardinalità di X).*

Dimostrazione. Supponiamo che $X = \{x_1, x_2, \ldots, x_k\}$. Sia $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$ un DFA per L. Sia $\forall i \in 1, \ldots, k \mid p_i = \delta(q_0, x_i)$. Se $\#Q < k$ allora esistono due stati uguali

$$\exists i, j \in 1, \ldots, k \mid i \neq j \land p_1 = p_j$$

Ma x_i, x_j sono distinguibili, quindi

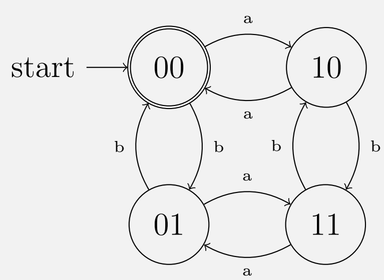
$$\exists z \mid (xz \in L \land yz \notin L) \lor (xz \notin L \land yz \in L)$$

Ma questo z non può esistere, quindi la $\#Q$ deve per forza essere $\leq k$. □

Questo può essere anche utile per determinare se un linguaggio non può essere definito da un automa a stati finiti. Infatti se si trova un insieme X infinito, allora il linguaggio non può essere regolare.

Sia $\Sigma = \{a, b\}$ e

$$L = \{x \in \Sigma^* \mid \#_a(x) \text{ è pari } \wedge \#_b(x) \text{ è pari } \}$$



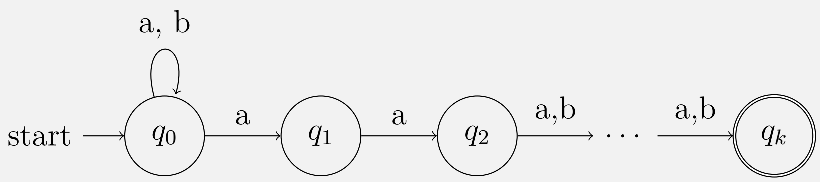
Costruisco $X = \{\epsilon, a, b, ab, \}$, per vedere che sono distinguibili costruiamo

z	ϵ	a	b	ab
ϵ		ϵ	ϵ	ϵ
a			a	a
b				b
ab				

Dato il linguaggio

$$\mathcal{L}_n = \{x \in \{a, b\}^* \mid \text{l'}n\text{-esimo simbolo da destra di } x \text{ è una } a\}$$

L'automa si ricorda gli ultimi n simboli, quindi ha 2^n stati. Mentre l'NFA corrispondente è



questo ha $n + 1$ stati.
Possiamo fare meglio per il deterministico? Costruiamo $X = \{a, b\}^n = \{x \in \{a, n\}^* \mid |x| = n\}$. Siano $x, y \in X$, con

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

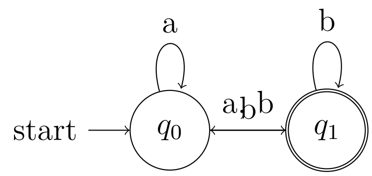
Visto che $x \neq y$, $\exists i \mid x_i \neq y_i$. Supponiamo che $x_i = a$, ed $y_i = b$, quindi

$$x = x_1 x_2 \ldots a \ldots x_n$$
$$y = y_1 y_2 \ldots b \ldots y_n$$

Ora per distinguere le tue parole basta scegliere $z = \{a, b\}^{i-1}$, ad esempio $z = a^{i-1}$, con questa z $xz \in \mathcal{L}$ e $yz \notin \mathcal{L}$. Visto che la cardinalità di X è di 2^n allora non possiamo costruire un automa di meno di 2^n stati.
Questo mostra come gli automi non deterministici possano essere molto più compatti.

Se definiamo $\mathcal{L} = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ ha X infinito

1.1 Costruzione coi sottoinsiemi



$\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ NFA definisco $\mathcal{A}' = \langle Q', \Sigma, \delta', q_0', F' \rangle$, con $Q' = 2^Q$ e $q_0' = \{q_0\}$. Quindi

$$\forall a \in Q', a \in \Sigma \mid \delta'(\alpha, a) = \bigcup_{q \in \alpha} \delta(q, a)$$

ed

$$F' = \{\alpha \in Q' \mid \alpha \cap F \neq \varnothing\}$$

Se l'NFA ha n stati, il DFA ottenuto ha 2^n stati.

Costruiamo ora l'automa M_n, o automa di Meyer&Fisher (1971), non deterministico tale per cui l'automa deterministico corrispondente ha necessariamente 2^n stati. Quindi è l'automa $\mathcal{A} = \langle Q = \{0, 1, \dots, n-1\}, \{a, b\}, \delta, q_0 = 0, F = \{0\} \rangle$ con δ definita come

$$\delta(i, a) = (i + 1) \mod n$$
$$\delta(i, b) = \begin{cases} 0 & \text{se } i = 0 \\ \{i, 0\} & \text{altrimenti} \end{cases}$$

Sia $S \subseteq \{0, \dots, n-1\}$, definisco w_s come

$$w_s = \begin{cases} b & \text{se } S = \varnothing \\ a^i & \text{se } S = \{i\} \\ a^{e_k - e_{k-1}} b a^{e_{k-1} - e_{k-2}} b \dots b a^{e_2 - e_1} b a^{e_1} & \text{se } S = \{e_1, e_2, \dots, e_k\} \text{ con } k \geq 2, e_1 < e_2 < \dots < e_k \end{cases}$$

Ad esempio prendiamo $S = \{1, 3, 4\}$, la w_S corrispondente è

$$w_S = aba^2ba = abaaba$$

Mostriamo che l'insieme di stati raggiunto da questa stringa è esattamente $\{1, 3, 4\} = S$. Mostriamo anche per $S = \{0, 2\}$, $w_S = a^2b$. O ancora $S = \varnothing$, $w_S = b$.

Proprietà 1. *È possibile dimostrare che* $\forall S \subseteq \{0, \dots, n-1\} \mid \delta(0, w_S) = S$.

Proprietà 1

Proprietà 2. *Siano* $S, T \subseteq \{0, \dots, n-1\}$, *se* $S \neq T$ *allora* w_S *e* w_T *sono distinguibili.*

Proprietà 2

Dimostrazione. Se $S \neq T$ esiste un numero che appartiene a S ma non a T, chiamiamolo $x \in S \; T$. Abbiamo che

Complemento, non ricordo il simbolo

$$\delta(0, w_S) = S$$
$$\delta(0, w_T) = T$$

Dato uno stato y ci si mettono a^{n-y} per arrivare allo stato finale. Quindi da w^S ci si mettono a^{n-x} per arrivare ad uno stato finale.

$$0 \overset{w_S}{\rightsquigarrow} x \overset{a^{n-x}}{\rightsquigarrow} 0$$

Sia $y \in T$, se $y \neq x$, allora

$$0 \overset{w_T}{\rightsquigarrow} y \overset{a^{n-x}}{\not\rightsquigarrow} 0$$

Allora a^{n-x} distingue S e T.

Ed abbiamo che $X = \{w_S | S \subseteq \{0, \dots, n-1\}\}$ è foramto da stringhe a coppie di stringhe Quindi ogni DFA ha almeno 2^n stati. $\square$

Questo è il linguaggio (più o meno) delle stringhe che hanno un suffisso della forma bx dove $\#_a(x)$ è multiplo di n.

1.2 ϵ-mosse

Possiamo, in certi casi, introdurre delle transizioni sulle parole vuote.

> Supponiamo di volere l'automa che riconosce i numeri con segno (opzionale) Oppure supponiamo di volere l'automa per $a^l b^m c^n$

Generalmente se ho

$$q_0 \xrightarrow{\epsilon} q_1 \xrightarrow{a} q_2 \xrightarrow{\epsilon} q_3$$

diventa

$$q \xrightarrow{\epsilon} q_3$$

Inoltre se uno stato ha un cammino formato da ϵ-mosse fino ad uno stato finale, esso stesso è finale.

Questo necessariamente introduce non determinismo.

1.3 Stati iniziali multipli

Questa è un'altra estensione che genera non determinismo. Per renderli deterministici si crea lo stato iniziale che è l'insieme dei diversi stati iniziali.

Se si prende un automa con stati iniziali multipli, ma transizioni deterministiche, si può comunque avere un gap esponenziale tra il numero di stati del'NFA e del DFA.