



SAPIENZA
UNIVERSITÀ DI ROMA

**Facoltà di Ingegneria dell'Informazione, Informatica e
Statistica
Dipartimento di Informatica**

Automi Calcolabilità e Complessità

Autore:
Simone Lidonnici

25 settembre 2024

Indice

1	Linguaggi regolari	1
1.1	Automi Deterministici a Stati Finiti (DFA)	1
1.1.1	Configurazione di una DFA	3
1.2	Linguaggi regolari	4
1.2.1	Proprietà dei linguaggi regolari	4
E	Esercizi	5
E.1	Esercizi sui linguaggi regolari	5
E.1.1	Costruire un DFA da un linguaggio	5

1

Linguaggi regolari

Definizione di linguaggio

Dato un **alfabeto** Σ , cioè un insieme di elementi, un **linguaggio** Σ^* è l'insieme di tutte le stringhe ottenibili usando l'alfabeto Σ .

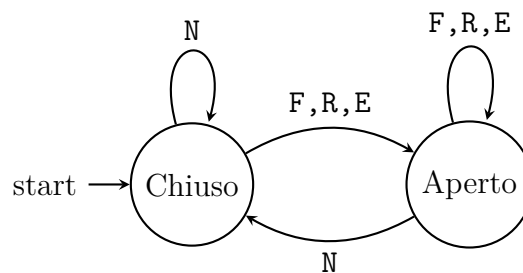
1.1 Automi Deterministici a Stati Finiti (DFA)

Il modello usato per definire i **linguaggi regolari** è l'**automa a stati finiti**, cioè una macchina che permette tramite l'input di passare da uno stato ad un altro, che ha memoria limitata e gestione dell'input limitata, ma è molto semplice.

Esempio:

Una porta che si apre tramite dei sensori può essere descritta tramite un automa con due stati (Aperta e Chiusa) e quattro input dati dai sensori (N, F, R, E):

- N: se non ci sono persone da nessun lato della porta
- F: se c'è una persona davanti alla porta
- R: se c'è una persona dietro la porta
- E: se ci sono persone da entrambi i lati della porta



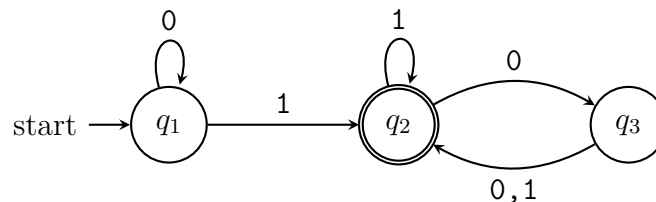
Automa Deterministico a Stati Finiti (DFA)

Un **DFA** (**D**eterministic **F**inite **A**utomaton) è una tupla $(Q, \Sigma, \delta, q_0, F)$ in cui:

- Q è l'insieme degli stati dell'automa
- Σ è l'alfabeto dell'automa
- $\delta : Q \times \Sigma \rightarrow Q$ è la funzione di transizione degli stati
- $q_0 \in Q$ è lo stato iniziale dell'automa
- $F \subseteq Q$ è l'insieme di **stati accettanti** dell'automa

Esempio:

Preso il seguente DFA:



In questo caso:

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $\delta = \begin{array}{c|ccc} \delta & q_1 & q_2 & q_3 \\ \hline 0 & q_1 & q_3 & q_2 \\ 1 & q_2 & q_2 & q_2 \end{array}$
- q_1 è lo stato iniziale dell'automa
- $F = \{q_2\}$ è l'insieme degli stati accettanti

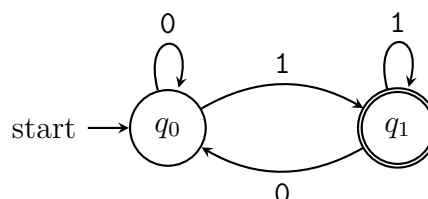
Funzione di transizione estesa

Dato un DFA D , definiamo una **funzione di transizione estesa** $\delta^* : Q \times \Sigma^* \rightarrow Q$ in modo ricorsivo:

$$\begin{cases} \delta^*(q, \varepsilon) = \delta(q, \varepsilon) = q \\ \delta^*(q, ax) = \delta^*(\delta(q, a), x) & a \in \Sigma, x \in \Sigma^* \end{cases}$$

Esempio:

Preso il seguente DFA:



In questo DFA:

- $\delta^*(q_0, 011) = \delta^*(\delta(q_0, 0), 11) = \delta^*(q_0, 11) = \delta^*(\delta(q_0, 1), 1) = \delta^*(q_1, 1) = \delta^*(\delta(q_1, 1), \varepsilon) = \delta^*(q_1, \varepsilon) = q_1$

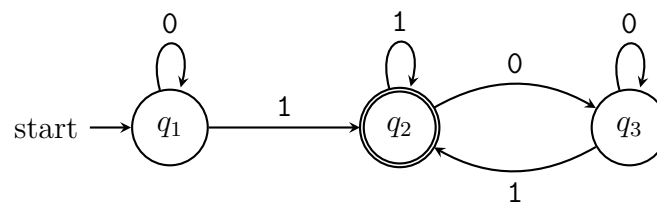
Linguaggio di un automa

Il **linguaggio di un DFA** D è l'insieme delle stringhe in input che l'automa accetta, cioè quelle per cui l'automa termina in uno stato accettante $q_0 \in F$.

Ogni automa ha un solo linguaggio che si scrive $L(D) = \{x \in \Sigma^* \mid D \text{ accetta } x\}$. Una stringa $x \in \Sigma^*$ sarà accettata da una DFA se $\delta^*(q_0, x) = q \in F$

Esempio:

Preso il seguente DFA:



Il linguaggio di questa DFA è l'insieme di tutte le stringhe che finiscono con 1:

$$L(D) = \{x \in \{0, 1\}^* \mid x = y1 \wedge y \in \{0, 1\}^*\}$$

1.1.1 Configurazione di una DFA

Configurazione di una DFA

Dato un DFA D , una **configurazione** di D è una coppia $Q \times \Sigma^*$ che indica:

- lo stato attuale dell'automa
- l'input ancora da leggere

La configurazione iniziale è sempre (q_0, x) .

Passo di computazione

Un **passo di computazione** è una relazione binaria con simbolo \vdash_D per cui:

$$(q_1, ax) \vdash_D (q_2, x) \iff \delta(q_1, a) = q_2$$

La **chiusura per riflessione e transitività** di \vdash_D , scritta come \vdash_D^* , ha delle proprietà:

1. $(q_1, ax) \vdash_D (q_2, x) \implies (q_1, ax) \vdash_D^* (q_2, x)$
2. $\forall q, x (q, x) \vdash_D^* (q, x)$
3. $(q_1, abc) \vdash_D (q_2, bc) \vdash_D (q_3, c) \implies (q_1, abc) \vdash_D^* (q_3, c)$

Una stringa $x \in \Sigma^*$ sarà accettata da un DFA se:

$$\exists q \in F (q_0, x) \vdash_D^* (q, \varepsilon)$$

1.2 Linguaggi regolari

Insieme dei Linguaggi regolari

Dato un alfabeto Σ , l'insieme dei **linguaggi regolari** di Σ , scritto come REG, è l'insieme dei linguaggi per cui esiste una DFA che li accetta:

$$\text{REG} = \{L \subseteq \Sigma^* \mid \exists D L(D) = L\}$$

1.2.1 Proprietà dei linguaggi regolari

I linguaggi sono insiemi di stringhe di un alfabeto Σ , quindi dati due linguaggi $L_1, L_2 \subseteq \Sigma^*$ possiamo definire le operazioni:

- **Unione:**

$$L_1 \cup L_2 = \{x \in \Sigma^* \mid x \in L_1 \vee x \in L_2\}$$

- **Intersezione:**

$$L_1 \cap L_2 = \{x \in \Sigma^* \mid x \in L_1 \wedge x \in L_2\}$$

- **Complemento:**

$$\neg L = \{x \in \Sigma^* \mid x \notin L\}$$

- **Concatenazione:**

$$L_1 \circ L_2 = \{xy \in \Sigma^* \mid x \in L_1 \wedge y \in L_2\}$$

- **Potenza:**

$$L^n = \begin{cases} \{\varepsilon\} & n = 0 \\ L \circ L^{n-1} & n > 0 \end{cases}$$

- **Star di Kleene:**

$$L^* = \{x_1 \dots x_k \in \Sigma^* \mid x_i \in L\} = \bigcup_{n \geq 0} L^n$$

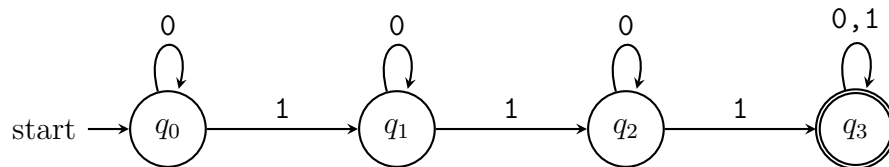
E

Esercizi

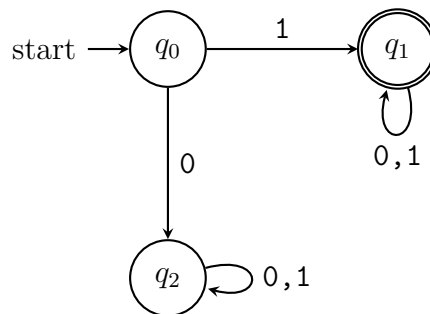
E.1 Esercizi sui linguaggi regolari

E.1.1 Costruire un DFA da un linguaggio

1. Dato un linguaggio $L(D) = \{x \in \{0, 1\}^* | w_H(x) \geq 3\}$, per cui $w_H(x)$ = numero di 1 in x , costruire un DFA che accetta questo linguaggio.



2. Dato un linguaggio $L(D) = \{x \in \{0, 1\}^* | x = 1y \wedge y \in 0, 1^*\}$, costruire un DFA che accetta questo linguaggio.



3. Dato un linguaggio $L(D) = \{x \in \{0, 1\}^* | x = 0^n 1\}$, costruire un DFA che accetta questo linguaggio.

