"Sapienza" Università di Roma
Ingegneria dell'Informazione,
Informatica e Statistica
Dipartimento di Informatica

Automi: Calcolabilità e Complessità

Appunti integrati con il libro "Introduzione alla teoria della computazione",
Michael Sipser

Author
Alessio Bandiera

2 settembre 2023

Indice

Informazioni e Contatti

Prerequisiti consigliati:

- TODO: DA DECIDERE

Segnalazione errori ed eventuali migliorie:

Per segnalare eventuali errori e/o migliorie possibili, si prega di utilizzare il **sistema di Issues fornito da GitHub** all'interno della pagina della repository stessa contenente questi ed altri appunti (link fornito al di sotto), utilizzando uno dei template già forniti compilando direttamente i campi richiesti.

Gli appunti sono in continuo aggiornamento, pertanto, previa segnalazione, si prega di controllare se l'errore sia ancora presente nella versione più recente.

Licenza di distribuzione:

Contatti dell'autore e ulteriori link:

- Github: **https://github.com/ph04**
- Email: **alessio.bandiera02@gmail.com**
- LinkedIn: **Alessio Bandiera**

1

Linguaggi regolari

1.1 Automi finiti

1.1.1 Linguaggi

Definizione 1.1.1.1: Alfabeto

Si definisce **alfabeto** un qualsiasi insieme finito, non vuoto; i suoi elementi sono detti **simboli**.

Esempio 1.1.1.1 (Alfabeto)**.** $\Sigma = \{\mathtt{0}, \mathtt{1}, \mathtt{x}, \mathtt{y}, \mathtt{z}\}$ è un alfabeto, composto da 5 simboli.

Definizione 1.1.1.2: Stringa

Sia Σ un alfabeto; una **stringa su** Σ è una sequenza finita di simboli di Σ; la **stringa vuota** appartiene ad ogni alfabeto, ed è denotata con ε.

- Data una stringa w di Σ, allora $|w|$ è la lunghezza di w.
- Se w ha lunghezza $n \in \mathbb{N}$, allora è possibile scrivere che $w = w_1 w_2 \cdots w_n$ con $w_i \in \Sigma$ e $i \in [1, n]$.

Esempio 1.1.1.2 (Stringa)**.** Sia $\Sigma = \{\mathtt{0}, \mathtt{1}, \mathtt{x}, \mathtt{y}, \mathtt{z}\}$ un alfabeto; allora una sua possibile stringa è $w = \mathtt{x1y0z}$.

Definizione 1.1.1.3: Stringa inversa

Sia Σ un alfabeto, e $w = w_1 w_2 \cdots w_n$ una sua stringa; allora si definisce l'**inversa** di w come segue: $w^{\mathcal{R}} = w_n w_{n-1} \cdots w_1$

Definizione 1.1.1.4: Concatenazione

Sia Σ un alfabeto, e $x = x_1x_2 \cdots x_n, y = y_1y_2 \cdots y_n$ due sue stringhe; allora xy è la stringa ottenuta attraverso la **concatenazione** di x ed y.

Per indicare una stringa concatenata con se stessa k volte, si utilizza la notazione $x^k = \underbrace{xx \cdots x}_{k}$.

Definizione 1.1.1.5: Prefisso

Sia Σ un alfabeto, ed x, y due sue stringhe; allora x è detto essere un **prefisso** di y, se $\exists z \mid xz = y$, con z stringa in Σ.

Esempio 1.1.1.3 (Prefisso)**.** Sia $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ un alfabeto; allora la stringa $x = \mathtt{ab}$ è prefisso della stringa $y = \mathtt{abc}$, poiché esiste una stringa $z = \mathtt{c}$ tale per cui $xz = y$.

Definizione 1.1.1.6: Linguaggio

Sia Σ un alfabeto; si definisce **linguaggio** un insieme di stringhe di Σ. Un linguaggio è detto **prefisso**, se nessun suo elemento è prefisso di un altro.

1.1.2 Automi finiti

Definizione 1.1.2.1: Automa finito

Un **automa finito**, o **macchina**, è una quintupla $(Q, \Sigma, \delta, q_0, F)$, dove

- Q è l'**insieme degli stati** dell'automa, un insieme *finito*
- Σ è l'**alfabeto** dell'automa, un insieme *finito*
- $\delta : Q \times \Sigma \to Q$ è la **funzione di transizione**, che definisce la relazione tra gli stati
- $q_0 \in Q$ è lo **stato iniziale**
- $F \subseteq Q$ è l'**insieme degli stati accettanti**, sui quali le stringhe possono terminare

Esempio 1.1.2.1 (Automa finito)**.** Un esempio di automa finito è il seguente:

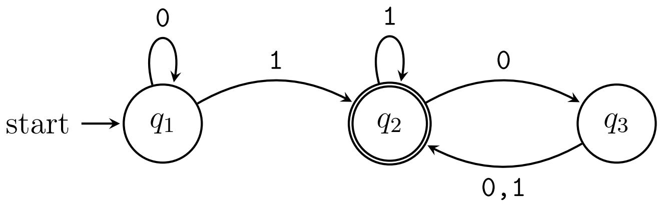

Figura 1.1: Un automa finito.

esso può essere descritto secondo la quintupla $(Q, \Sigma, \delta, q_0, F)$ come segue:

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{\texttt{0}, \texttt{1}\}$
- δ è la seguente:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

- q_1 è lo stato iniziale
- $F = \{q_2\} \subseteq Q$

Definizione 1.1.2.2: Linguaggio di un automa

Sia M un automa; allora il **linguaggio** di M è un insieme $L(M)$ contenente tutte le stringhe accettate da M; simmetricamente, si dice che M **riconosce** $L(M)$.

Esempio 1.1.2.2 (Linguaggio di un automa)**.** Si consideri il seguente automa M_1:

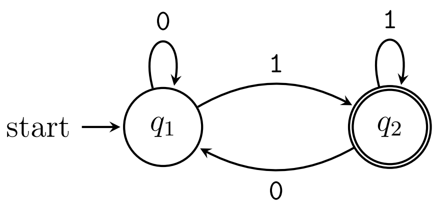

Figura 1.2: Un automa M_1.

sapendo che $\Sigma = \{\texttt{0}, \texttt{1}\}$, che q_1 è lo stato iniziale, e che $F = \{q_2\}$, ci si convince facilmente che

$$L(M_1) = \{w \mid w = w_1 w_2 \cdots w_{n-1}\texttt{1}, n \in \mathbb{N}\}$$

ovvero, M_1 accetta tutte e sole le stringhe che terminano per $\texttt{1}$.

Definizione 1.1.2.3: Stringhe accettate

Sia $M = (Q, \Sigma, \delta, q_0, F)$ un automa, e sia $w = w_1 \cdots w_n$ una stringa tale per cui $\forall i \in [1, n] \quad w_i \in \Sigma$; allora, M **accetta** w se esiste una sequenza di stati $r_0, \ldots, r_n \in Q$ tali per cui

- $r_0 = q_0$
- $\forall i \in [1, n-1] \quad \delta(r_i, w_{i+1}) = r_{i+1}$
- $r_n \in F$

Dato un linguaggio A, si ha che M **riconosce** A se e solo se $A = \{w \mid M \text{ accetta } w\}$.

Definizione 1.1.2.4: Linguaggio regolare

Un linguaggio è detto **regolare** se e solo se esiste un automa finito che lo riconosce.

1.2 Operazioni regolari

1.2.1 Definizioni

Definizione 1.2.1.1: Unione

Siano A e B due linguaggi; allora, si definisce l'**unione** di A e B il seguente linguaggio:

$$A \cup B = \{x \mid x \in A \lor x \in B\}$$

Definizione 1.2.1.2: Concatenazione

Siano A e B due linguaggio; allora, si definisce la **concatenazione** di A e B il seguente linguaggio:

$$A \circ B = \{xy \mid x \in A \land y \in B\}$$

Definizione 1.2.1.3: Star

Sia A un linguaggio; allora, si definisce l'operazione unaria **star** che definisce il seguente linguaggio:

$$A^* = \{x_1 x_2 \cdots x_k \mid k \geq 0 \land \forall i \in [1, k] \quad x_i \in A\}$$

Si noti che $k = 0 \implies \varepsilon \in A^*$ per ogni linguaggio A.

Esempio 1.2.1.1 (Operazioni regolari)**.** Sia $\Sigma = \{\mathtt{a}, \ldots, \mathtt{z}\}$ l'alfabeto composto da 26 lettere, e siano $A = \{\mathtt{uno}, \mathtt{due}\}$ e $B = \{\mathtt{tre}, \mathtt{quattro}\}$ due linguaggi su Σ. Allora, si ha che

- $A \cup B = \{\texttt{uno}, \texttt{due}, \texttt{tre}, \texttt{quattro}\}$
- $A \circ B = \{\texttt{unotre}, \texttt{unoquattro}, \texttt{duetre}, \texttt{duequattro}\}$
- $A^* = \{\varepsilon, \texttt{uno}, \texttt{due}, \texttt{unouno}, \texttt{unodue}, \texttt{dueuno}, \texttt{duedue}, \ldots\}$

Proposizione 1.2.1.1: Chiusura dell'unione

Siano A e B due linguaggi regolari su un alfabeto Σ; allora $A \cup B$ è regolare.

Dimostrazione. Per definizione, A e B sono linguaggi regolari, dunque esistono due automi finiti

$$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$$
$$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$$

tali da riconoscere rispettivamente A e B. Allora, sia $M = (Q, \Sigma, \delta, q_0, F)$ l'automa finito definito come segue:

- $Q := Q_1 \times Q_2 = \{(r_1, r_2) \mid r_1 \in Q_1 \land r_2 \in Q_2\}$, scelto tale in quanto permette di avere tutte le possibili combinazioni di stati dei due automi di partenza;
- $\forall (r_1, r_2) \in Q, a \in \Sigma \quad \delta((r_1, r_2), a) := (\delta_1(r_1, a), \delta_2(r_2, a))$, scelta tale in quanto permette di simulare entrambi gli automi di partenza contemporaneamente, mandando ogni stato di M_1 ed M_2 dove sarebbero andati nei rispettivi automi di appartenenza;
- $q_0 := (q_1, q_2)$, scelto tale in quanto deve essere lo stato in cui entrambe gli automi in ipotesi iniziavano;
- $F := (F_1 \times Q_2) \cup (Q_1 \times F_2) = \{(r_1, r_2) \mid r_1 \in F_1 \lor r_2 \in F_2\}$, scelto tale in quanto permette di simulare gli stati accettanti di entrambi gli automi, e vanno prese tutte le coppie che vedono almeno uno dei due stati come accettanti poiché altrimenti non si accetterebbero delle stringhe accettate da M_1 ed M_2 in partenza.

Allora, poiché M è in grado di simulare M_1 ed M_2 contemporaneamente, per costruzione accetterà ogni stringa di A e di B, dunque riconoscendo $A \cup B$, e dunque $A \cup B$ è regolare per definizione. $\square$