

# Automi e Linguaggi Formali

## Parte 1 – Linguaggi regolari e automi a stati finiti

Davide Bresolin  
Ultimo aggiornamento: 28 febbraio 2024



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

- 1 Introduzione al corso
- 2 Organizzazione del Corso
- 3 Automi a Stati Finiti Deterministici

## Un Informatico:

- come un **matematico**, usa un linguaggio rigoroso per descrivere le cose
- come un **ingegnere**, progetta sistemi complessi
- come uno **scienziato**, osserva il comportamento dei sistemi, formula ipotesi, e ne verifica i risultati

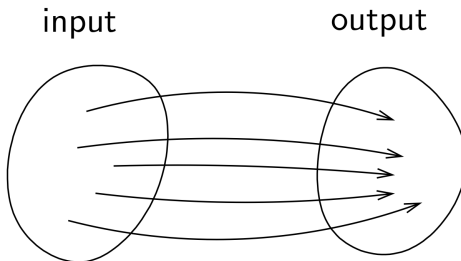
## In questo corso faremo i *matematici* e gli *scienziati*:

- vedremo degli strumenti per **descrivere** e **risolvere** problemi,
- ne studieremo le **proprietà**,
- **confronteremo** i diversi strumenti,
- per stabilire **cosa possono fare** e cosa no

## Problema

Per descrivere un **problema** dobbiamo specificare:

- l'insieme dei possibili input
- l'insieme dei possibili output
- la relazione tra input e output



- **Algoritmo** – procedura meccanica che esegue delle computazioni (e può essere eseguita da un calcolatore)
- Un algoritmo **risolve** un dato problema se:
  - Per ogni input, il calcolo dell'algoritmo si interrompe dopo un numero finito di passaggi.
  - Per ogni input, l'algoritmo produce un output corretto.
- **Correttezza** di un algoritmo – verificare che l'algoritmo risolva realmente il problema dato
- **Complessità computazionale** di un algoritmo:
  - **complessità temporale** – come varia il tempo di esecuzione rispetto alla dimensione dei dati di input
  - **complessità spaziale** – come varia la quantità di memoria utilizzata rispetto alla dimensione dei dati di input

## Linguaggi Formali

- Astrazione della nozione di problema
- I problemi possono essere espressi come **linguaggi** (= insiemi di stringhe)
  - Le soluzioni determinano se una determinata stringa è nell'insieme o no
    - ad esempio: un certo intero  $n$  è un numero primo?
- Oppure, come **trasformazioni tra linguaggi**
  - Le soluzioni trasformano la stringa di input in una stringa di output
    - ad esempio: quanto fa  $3 + 5$ ?

## Linguaggi Formali

- Quindi in sostanza tutti i processi computazionali possono essere ridotti ad uno tra:
  - Determinazione dell'**appartenenza** a un insieme (di stringhe)
  - **Mappatura** tra insiemi (di stringhe)
- Formalizzeremo il concetto di computazione meccanica:
  - dando una definizione precisa del termine “algoritmo”
  - caratterizzando i problemi che sono o non sono adatti per essere risolti da un calcolatore.

## Automi

- Gli **automati** (singolare automa) sono dispositivi matematici astratti che possono:
  - determinare l'appartenenza di una stringa ad un insieme di stringhe
  - trasformare una stringa in un'altra stringa
- Hanno tutti gli **aspetti** di un computer:
  - input e output
  - memoria
  - capacità di prendere decisioni
  - trasformare l'input in output



## Automi

- Il tipo di **memoria** è cruciale:
  - memoria finita
  - memoria infinita:
    - con accesso limitato
    - con accesso illimitato
- Abbiamo diversi tipi di automi per diversi classi di linguaggi
- I diversi tipi di automi si differenziano per
  - la quantità di memoria (finita vs infinita)
  - il tipo di accesso alla memoria (limitato vs illimitato)

- 1 Introduzione al corso
- 2 Organizzazione del Corso
- 3 Automi a Stati Finiti Deterministici

**Docente:** Davide Bresolin

**e-mail:** `davide.bresolin@unipd.it`

**ufficio:** Stanza 3DA7, III Piano, corridoio A-D della Torre  
Archimede, Dipartimento di Matematica, via Trieste

**ricevimento:** **su appuntamento**

- **Parte 1:** linguaggi regolari
  - automi a stati finiti
  - espressioni e linguaggi regolari
- **Parte 2:** linguaggi liberi da contesto
  - grammatiche e linguaggi liberi dal contesto
  - automi a pila
- **Parte 3:** indecidibilità e intrattabilità
  - macchine di Turing
  - concetto di indecidibilità
  - problemi intrattabili
  - classi P e NP



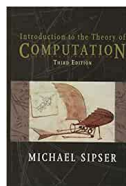
*M. Sipser*

Introduzione alla teoria della computazione

*M. Sipser*

Introduction to the theory of computation

Va bene **qualsiasi edizione** (1a, 2a, 3a)



- Vi si accede da <https://stem.elearning.unipd.it>
- Autenticazione tramite le proprie credenziali UniPD
- Pubblicazione di slide e altro materiale del corso
- Esercizi e soluzioni
- Comunicazioni e aggiornamenti

**Tutor:** Gabriel Rovesti  
email: [gabriel.rovesti@studenti.unipd.it](mailto:gabriel.rovesti@studenti.unipd.it)

**Incontri:** tutti i **venerdì**, a partire dall'**8 marzo**  
**Orario provvisorio:** aula e date/orari definitivi verranno comunicati in seguito.

- **Esercizi:** esercizi sul Moodle + esercizi pubblicati su Automata Tutor + attività svolte in aula.
- **Esame:** due modalità:
  - Due prove intermedie durante il corso
  - Esame scritto su tutto il programma



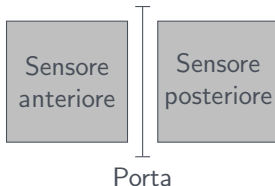
- Due prove intermedie:
  - nella settimana di sospensione della didattica 15-19 Aprile
  - nella settimana 10-14 Giugno
  - Le prove **sostituiscono l'esame**
    - devono essere entrambi sufficienti
- Per gli **appelli di Giugno e Luglio**:
  - i voti delle prove intermedie rimangono validi
  - si può recuperare un compitino insufficiente o migliorare il voto
- Per gli appelli di **Settembre e Febbraio**:
  - i voti delle prove intermedie non sono più validi
  - si deve fare l'esame completo

- 1 Introduzione al corso
- 2 Organizzazione del Corso
- 3 Automi a Stati Finiti Deterministici**

- Sono il più semplice **modello computazionale**
- Dispongono di una quantità di memoria **finita**
- Gli automi a stati finiti sono usati come **modello** per:
  - Software per la progettazione di circuiti digitali
  - Analizzatori lessicali di un compilatore
  - Ricerca di parole chiave in un file o sul web
  - Software per verificare sistemi a stati finiti, come protocolli di comunicazione

Costruiamo un esempio di controllore di una **porta automatica**:

- La porta si apre quando una persona si avvicina
- Un sensore di fronte alla porta rileva la presenza della persona
- Un sensore sul retro della porta rileva quando la persona ha attraversato la porta e se c'è qualcuno dietro la porta

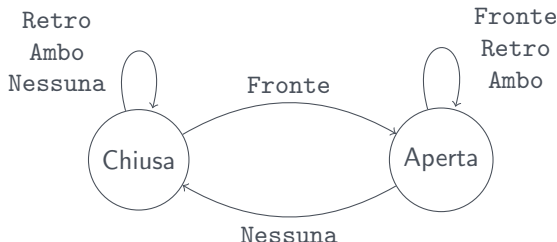


- La porta si può trovare in due stati: **Chiusa** o **Aperta**
- Ci sono quattro possibili input dai sensori:
  - **Fronte**: c'è una persona di fronte alla porta
  - **Retro**: c'è una persona dietro alla porta
  - **Ambo**: ci sono persone sia di fronte che dietro alla porta
  - **Nessuna**: non ci sono persone né davanti né dietro la porta

# Esempio: una porta automatica



- La porta si può trovare in due stati: **Chiusa** o **Aperta**
- Ci sono quattro possibili input dai sensori:
  - **Fronte**: c'è una persona di fronte alla porta
  - **Retro**: c'è una persona dietro alla porta
  - **Ambo**: ci sono persone sia di fronte che dietro alla porta
  - **Nessuna**: non ci sono persone né davanti né dietro la porta



Per rappresentare in maniera precisa l'esempio, dobbiamo definire alcuni concetti di base:

- Che cos'è un **alfabeto** (di simboli/messaggi/azioni)
- Che cos'è un **linguaggio formale**
- Che cos'è un **Automa a stati finiti deterministico**
- Cosa vuol dire che un automa **accetta** un linguaggio

**Alfabeto:** Insieme finito e non vuoto di simboli

- **Esempio:**  $\Sigma = \{0, 1\}$  alfabeto binario
- **Esempio:**  $\Sigma = \{a, b, c, \dots, z\}$  insieme di tutte le lettere minuscole
- **Esempio:** Insieme di tutti i caratteri ASCII

**Stringa:** (o **parola**) Sequenza finita di simboli da un alfabeto  $\Sigma$ , e.g. 0011001

**Stringa vuota:** La stringa con zero occorrenze di simboli da  $\Sigma$

- La stringa vuota è denotata con  $\varepsilon$

**Lunghezza di una stringa:** Numero di simboli nella stringa.

- $|w|$  denota la lunghezza della stringa  $w$
- $|0110| = 4$ ,  $|\varepsilon| = 0$



- **Potenze di un alfabeto:**  $\Sigma^k$  = insieme delle stringhe di lunghezza  $k$  con simboli da  $\Sigma$ 
  - Esempio:  $\Sigma = \{0, 1\}$

$$\Sigma^0 = \{\varepsilon\}$$

$$\Sigma^1 = \{0, 1\}$$

$$\Sigma^2 = \{00, 01, 10, 11\}$$

- Domanda: Quante stringhe ci sono in  $\Sigma^3$ ?
- L'insieme di **tutte le stringhe** su  $\Sigma$  è denotato da  $\Sigma^*$

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$$

- **Linguaggio:** dato un alfabeto  $\Sigma$ , chiamiamo linguaggio ogni sottoinsieme  $L \subseteq \Sigma^*$
- Esempi di linguaggi:
  - L'insieme delle parole italiane
  - L'insieme dei programmi C sintatticamente corretti
  - L'insieme delle stringhe costituite da  $n$  zeri seguiti da  $n$  uni:  
 $\{\varepsilon, 01, 0011, 000111, \dots\}$
  - Il **linguaggio vuoto**  $\emptyset$  non contiene nessuna parola
  - Il linguaggio che contiene solo la parola vuota:  
 $\{\varepsilon\}$
  - ...

Un Automa a Stati Finiti Deterministico (DFA) è una quintupla

$$A = (Q, \Sigma, \delta, q_0, F)$$

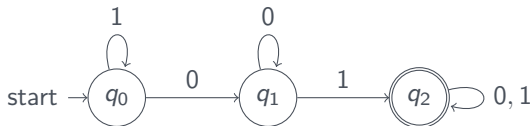
- $Q$  è un insieme finito di **stati**
- $\Sigma$  è un **alfabeto finito** (= simboli in input)
- $\delta : Q \times \Sigma \mapsto Q$  è una **funzione di transizione**
- $q_0 \in Q$  è lo **stato iniziale**
- $F \subseteq Q$  è un insieme di **stati finali**

Possiamo rappresentare gli automi sia come **diagramma di transizione** che come **tabella di transizione**.

**Esempio:** costruiamo un automa  $A$  che accetta il linguaggio delle stringhe con 01 come sottostringa

**Esempio:** costruiamo un automa  $A$  che accetta il linguaggio delle stringhe con 01 come sottostringa

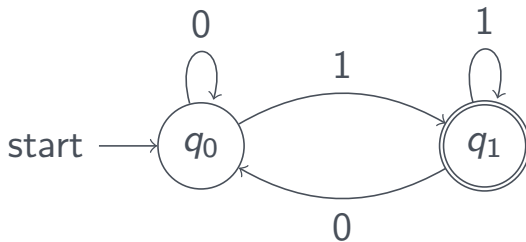
- L'automato come **diagramma di transizione**:



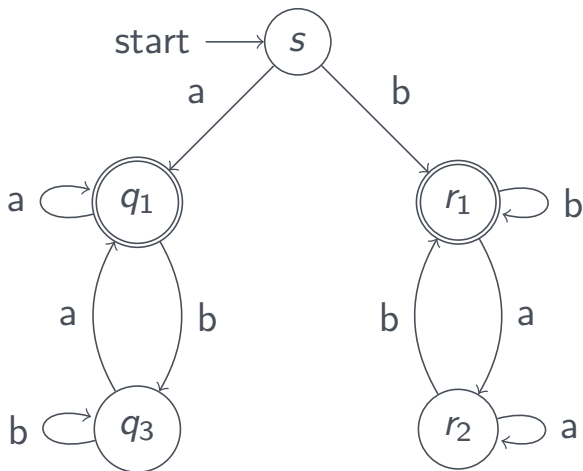
- L'automato come **tabella di transizione**:

	0	1
→ $q_0$	$q_1$	$q_0$
$q_1$	$q_1$	$q_2$
* $q_2$	$q_2$	$q_2$

# Cosa fa questo automa?



... e questo?



- Data una parola  $w = w_1 w_2 \dots w_n$ , la **computazione** dell'automa  $A$  con input  $w$  è una sequenza di stati  $r_0 r_1 \dots r_n$  che rispetta **due condizioni**:
  - 1  $r_0 = q_0$  (inizia dallo stato iniziale)
  - 2  $\delta(r_i, w_{i+1}) = r_{i+1}$  per ogni  $i = 0, \dots, n - 1$  (rispetta la funzione di transizione)
- Diciamo che la computazione **accetta** la parola  $w$  se:
  - 3  $r_n \in F$  (la computazione **termina in uno stato finale**)



- Un DFA  $A$  **accetta** la parola  $w$  se la computazione accetta  $w$
- Formalmente, il **linguaggio accettato** da  $A$  è

$$L(A) = \{w \in \Sigma^* \mid A \text{ accetta } w\}$$

- I linguaggi accettati da automi a stati finiti sono detti **linguaggi regolari**

DFA per i seguenti linguaggi sull'alfabeto  $\{0, 1\}$ :

- Insieme di tutte e sole le stringhe con un numero pari di zeri e un numero pari di uni
- Insieme di tutte le stringhe che finiscono con 00
- Insieme di tutte le stringhe che contengono esattamente tre zeri (anche non consecutivi)
- Insieme delle stringhe che cominciano o finiscono (o entrambe le cose) con 01