Automi e Linguaggi Formali

Parte 1 – Linguaggi regolari e automi a stati finiti

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Davide Bresolin
Ultimo aggiornamento: 24 febbraio 2025

Sommario

Pensare da Informatici

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

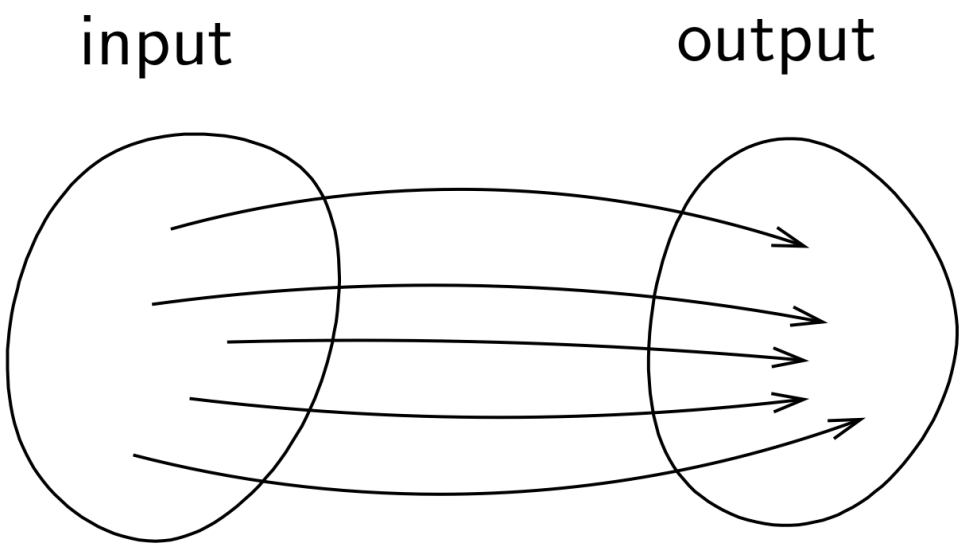
In questo corso faremo i *matematici* e gli *scienziati*:

- 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



Algoritmi e Problemi

- 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

Di cosa parla questo corso?

Linguaggi Formali

- Astrazione della nozione di problema
- I problemi possono sono 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 $3 + 5$?

Di cosa parla questo corso?

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.

Di cosa parla questo corso?

Automi

- Gli automi (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

Di cosa parla questo corso?

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)

Sommario

Docente del Corso

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

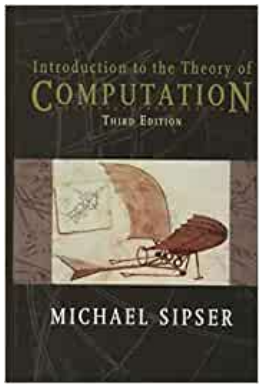
ricevimento: **su appuntamento**

Programma del Corso

- **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

Libro di testo

M. Sipser
Introduzione alla teoria della computazione



M. Sipser
Introduction to the theory of computation

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

Moodle del corso

- 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

Tutorato

Tutor: Gabriel Rovesti
email: gabriel.rovesti@studenti.unipd.it

Incontri: tutti i lunedì, a partire dal 10 marzo
Orario provvisorio: aula e date/orari definitivi verranno comunicati in seguito.

Esami, compitini ed esercizi

- **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

Prove intermedie

- Due prove intermedie:
  - nella settimana di sospensione della didattica 14-17 Aprile
  - nella settimana 9-12 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

Sommario

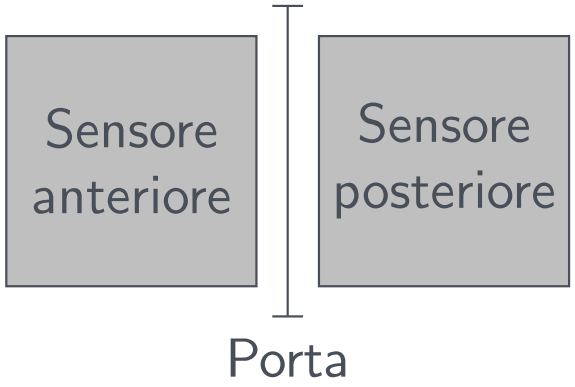
Gli Automi a Stati Finiti

- 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

Esempio: una porta automatica

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

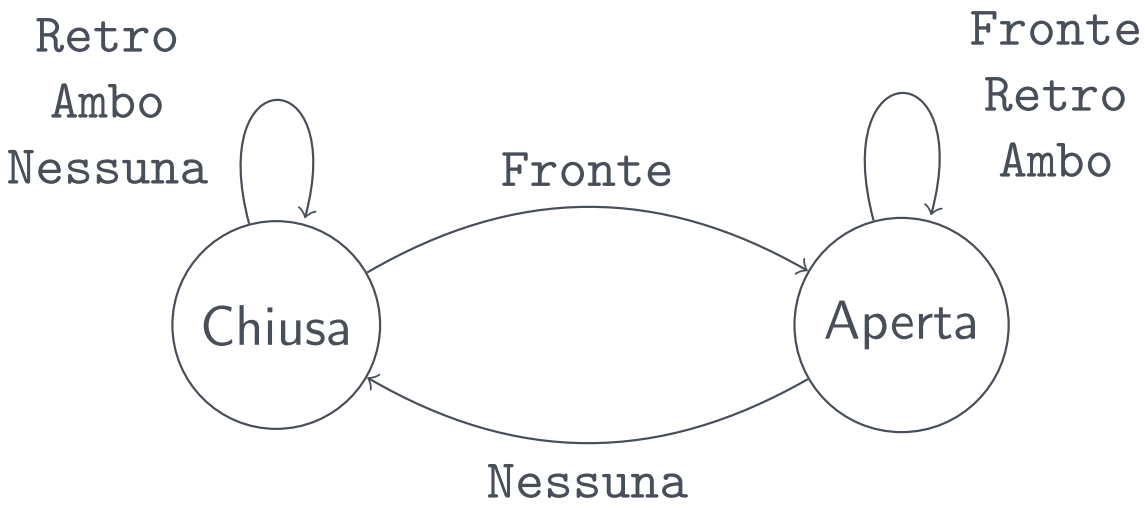


Esempio: una porta automatica

- 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

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



Alfabeti, linguaggi e automi a stati finiti

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

Alfabeti e stringhe

Alfabeto: Insieme finito e non vuoto di simboli

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

Stringa: (o parola) Sequenza finita di simboli da un alfabeto Σ, e.g. 0011001

Stringa vuota: La stringa con zero occorrenze di simboli da Σ

- La stringa vuota è denotata con ε

Lunghezza di una stringa: Numero di simboli nella stringa.

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

Potenze di un alfabeto

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

$$\Sigma^0 = \{\varepsilon\}$$
$$\Sigma^1 = \{0, 1\}$$
$$\Sigma^2 = \{00, 01, 10, 11\}$$

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

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

Linguaggi

- **Linguaggio:** dato un alfabeto Σ, 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 stringe 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:

    $$\{\varepsilon\}$$

  - ...

Automi a Stati Finiti Deterministici

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

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

- Q è un insieme finito di stati
- Σ 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.

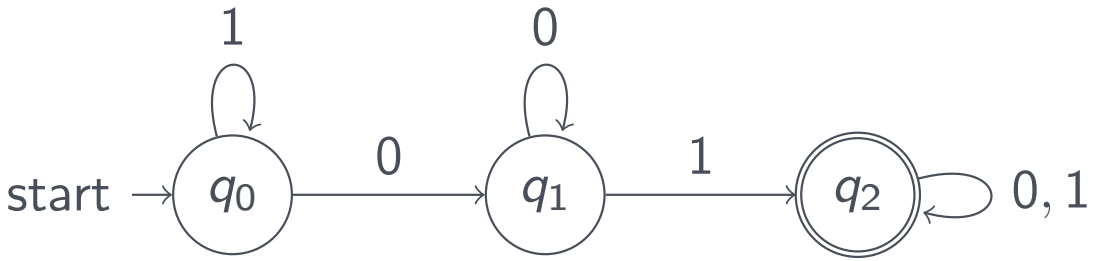
Diagrammi e tabelle di transizione

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

Diagrammi e tabelle di transizione

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

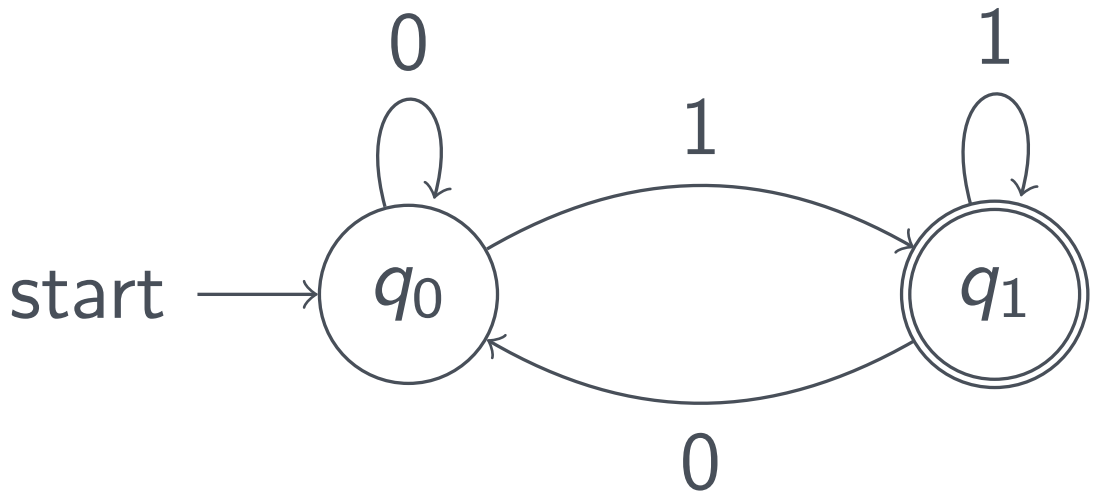
- L'automa come diagramma di transizione:



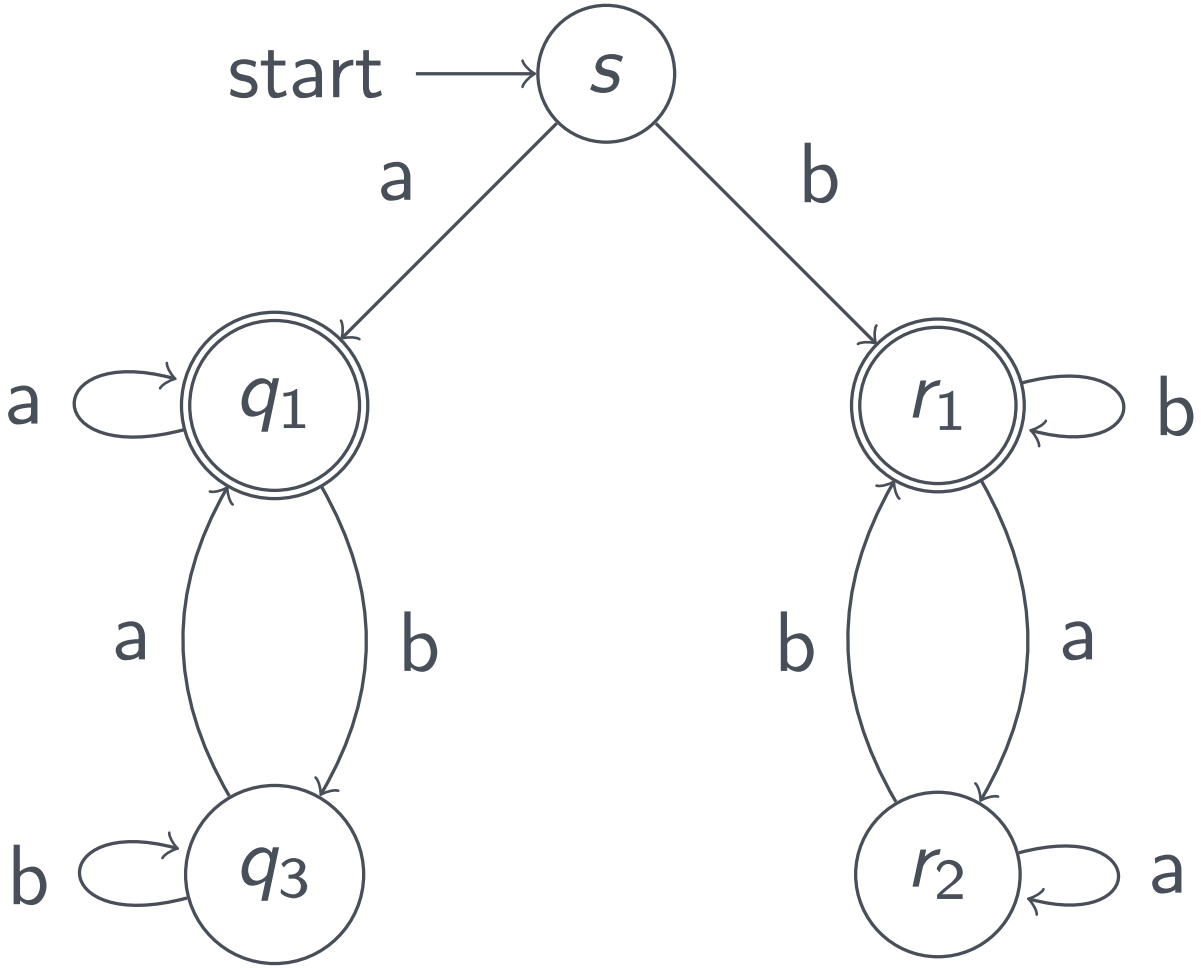
- L'automa come tabella di transizione:

	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$*q_2$	q_2	q_2

Cosa fa questo automa?



... e questo?



Computazione di un DFA

- 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)

Linguaggio accettato da un DFA

- 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

Esempi

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