

# Automi e Linguaggi Formali

a.a. 2016/2017

LT in Informatica  
27 Febbraio 2017



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Prima parte (4 settimane):

**Docente:** Davide Bresolin

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

**ufficio:** Stanza 501, V Piano, Scala A della Torre Archimede,  
Dipartimento di Matematica, via Trieste

**ricevimento:** lunedì 15:30-17:30

A seguire:

**Docenti:** Gilberto Filè e Lamberto Ballan

**I Settimana** Lun 27/2, 13:30–15:30, Aula LuM250  
Mar 28/2, 13:30–15:30, Aula LuM250  
Ven 3/3, 13:30–15:30, Aula LuM250

**II Settimana** Docente assente, **le lezioni sono sospese!**

**III Settimana** Lun 13/3, 13:30–15:30, Aula LuM250  
Mar 14/3, 13:30–15:30, Aula LuM250  
Ven 17/3, 13:30–15:30, Aula LuM250

**IV Settimana** Lun 20/3, 13:30–15:30, Aula LuM250  
Mar 21/3, 13:30–15:30, Aula LuM250  
Ven 24/3, 13:30–15:30, Aula LuM250

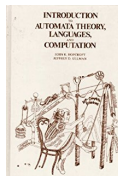
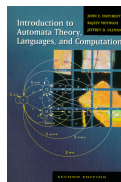
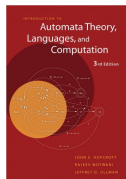
- **Parte 1:** linguaggi regolari
  - automi a stati finiti
  - espressioni e linguaggi regolari
- **Parte 2:** linguaggi liberi da contesto
  - grammatiche e linguaggi liberi dal contesto
  - automi a pila
- **Parte 3:** struttura dei compilatori e parsing
  - analisi lessicale
  - analisi sintattica: parsers top-down (LL) e bottom-up (LR)
- **Parte 4:** indecidibilità e intrattabilità
  - macchine di Turing
  - concetto di indecidibilità
  - problemi intrattabili
  - classi P e NP



*J. E. Hopcroft, R. Motwani, J. D. Ullman*  
**Automi, linguaggi e calcolabilità**

*J. E. Hopcroft, R. Motwani, J. D. Ullman*  
**Introduction to Automata Theory, Languages, and Computation**

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



- Accesso tramite le proprie credenziali UniPD
- Pubblicazione di slide e altro materiale del corso
- Esercizi e soluzioni
- Comunicazioni e aggiornamenti
- Forum di discussione

- **Esame:** Scritto e, se richiesto dai docenti, colloquio orale. Cinque appelli, tra Luglio, Settembre 2017 e Febbraio 2018.
- **Compitini (forse):** Due compitini, uno a meta' del corso e uno alla fine. Modalità da definire
- **Esercizi (prima parte del corso):** pubblicati il venerdì, corretti a lezione il venerdì successivo.

## Un Informatico:

- come un **matematico**, usa un linguaggio formale 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** un sistema,
- ne studieremo le **proprietà**,
- **confronteremo** i diversi strumenti,
- per stabilire **cosa possono fare** e cosa no



Gli automi a stati finiti sono usati come **modello** per:

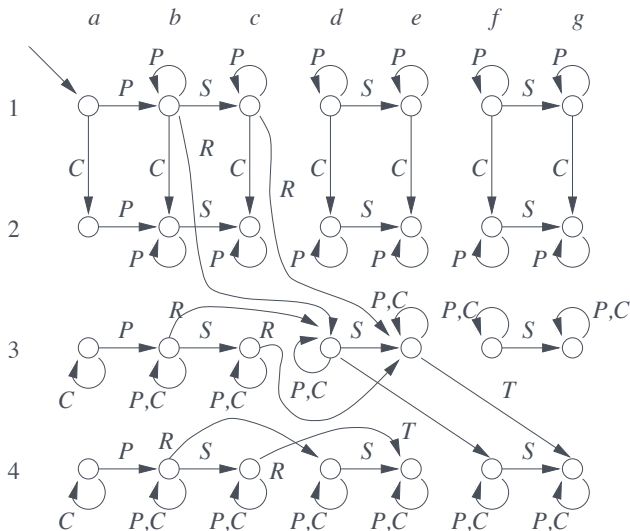
- Software per la progettazione di circuiti digitali
- Analizzatori lessicali di un compilatore
- Ricerca di parole chiave in un file o sul web
- Software per verificare sistemi a stati finiti, come protocolli di comunicazione

Costruiamo un esempio di **commercio elettronico**:

- Il **cliente** **paga** il negozio con moneta elettronica
- Il **cliente** può **cancellare** la moneta elettronica
- Il **negozio** riceve il pagamento e **spedisce** il prodotto al cliente
- Per completare il pagamento, il **negozio** **riscatta** la moneta elettronica
- La **banca** controlla la validità della moneta e **trasferisce** la somma al **negozio**

- Ogni automa reagisce solo ad alcuni messaggi:
  - Il cliente può ignorare riscatta e trasferisci
  - La banca può ignorare paga e spedisci
- Dobbiamo gestire anche comportamenti inattesi:
  - Cosa facciamo se cliente paga due volte?
- La definizione formale di automa prescrive che si debba reagire ad ogni messaggio
  - altrimenti il sistema “muore” e la computazione non prosegue
- Dobbiamo quindi aggiungere transizioni per avere una descrizione completa

# Il sistema completo



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  $\Sigma^*$ 
  - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$

- **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:  
 $\{\varepsilon, 01, 0011, 000111, \dots\}$
  - ...



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

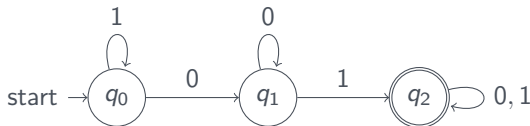
$$A = (Q, \Sigma, \delta, q_0, F)$$

- $Q$  è un insieme finito di **stati**
- $\Sigma$  è un **alfabeto finito** (= simboli in input)
- $\delta$  è una **funzione di transizione**  $(q, a) \mapsto q'$
- $q_0 \in Q$  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

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



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

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

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