

Automi e Linguaggi Formali

a.a. 2017/2018

LT in Informatica
26 Febbraio 2018



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Prima parte + Laboratorio + Terza parte

Docente: Davide Bresolin

e-mail: davide.bresolin@unipd.it

ufficio: Stanza 322, III Piano, Scala C della Torre Archimede,
Dipartimento di Matematica, via Trieste

ricevimento: martedì 16:30-18:30

Seconda parte + Terza parte

Docente: Gilberto Filè

- **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
- **Laboratorio:** due lezioni di esercitazione
 - costruzione di un parser per un linguaggio di programmazione
 - traduttore verso il linguaggio C / C++
 - **Giovedì 26 Aprile** e **giovedì 3 Maggio**, 12:30-14:30, LabP140
- **Parte 3:** indecidibilità e intrattabilità
 - macchine di Turing
 - concetto di indecidibilità
 - problemi intrattabili
 - classi P e NP

Calendario delle prime quattro settimane



I Settimana Lun 26/2, 12:30–14:30, Aula LuM250
Mar 27/2, 12:30–14:30, Aula LuM250
Gio 1/3, 12:30–14:30, Aula LuM250

II Settimana Lun 5/3, **le lezioni sono sospese!**
Mar 6/3, 12:30–14:30, Aula LuM250
Gio 8/3, 12:30–14:30, Aula LuM250

III Settimana Lun 12/3, 12:30–14:30, Aula LuM250
Mar 13/3, 12:30–14:30, Aula LuM250
Gio 15/3, 12:30–14:30, Aula LuM250

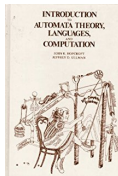
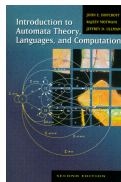
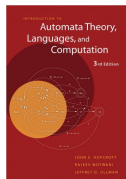
IV Settimana Lun 19/3, 12:30–14:30, Aula LuM250
Mar 20/3, 12:30–14:30, Aula LuM250
Gio 22/3, 12:30–14:30, Aula LuM250



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)



- Vi si accede da <https://elearning.unipd.it/math>
 - selezionando prima **Informatica – Triennale**
 - e poi **Automi e linguaggi formali 17/18**
- Autenticazione tramite le proprie credenziali UniPD
- Pubblicazione di slide e altro materiale del corso
- Esercizi e soluzioni
- Comunicazioni e aggiornamenti

- **Esame:** Scritto e, se richiesto dai docenti, colloquio orale.
Cinque appelli, tra Luglio, Settembre 2018 e Febbraio 2019.
- **Compitini:** Due compitini che sostituiscono l'esame (maggiori informazioni nella slide successiva!)
- **Esercizi (prima parte del corso):** test di autovalutazione sul Moodle + esercizi pubblicati il giovedì, corretti a lezione il lunedì successivo.

- **Due compitini:**
 - il primo durante la settimana di sospensione delle lezioni
 - 9–13 Aprile
 - il secondo alla fine del corso
- I compitini **sostituiscono l'esame**
 - devono essere entrambi sufficienti
- Per gli **appelli di Giugno e Luglio:**
 - compito diviso in due parti
 - si può recuperare un compitino insufficiente
- Dagli **appelli di Settembre** in poi:
 - si deve fare l'esame completo

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

- Un **quine** è un programma che **riproduce il suo stesso codice sorgente** senza usare funzioni di I/O
 - aprire il file sorgente e stampare il suo contenuto è “barare”!
- **Provate a scrivere un quine** nel vostro linguaggio di programmazione preferito
 - e inviate la soluzione sul Moodle!

- Un **quine** è un programma che **riproduce il suo stesso codice sorgente** senza usare funzioni di I/O
 - aprire il file sorgente e stampare il suo contenuto è “barare”!
- **Provate a scrivere un quine** nel vostro linguaggio di programmazione preferito
 - e inviate la soluzione sul Moodle!

Esempio di quine in Italiano:

Scrivi quanto segue due volte, la seconda tra virgolette.
“Scrivi quanto segue due volte, la seconda tra virgolette.”

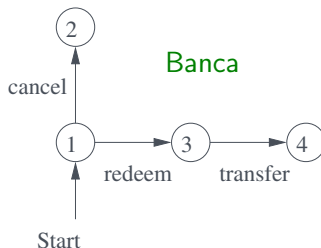
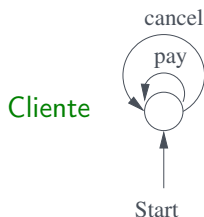
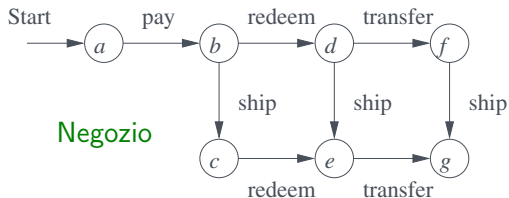
Gli automi a stati finiti sono usati come **modello** per:

- 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**:

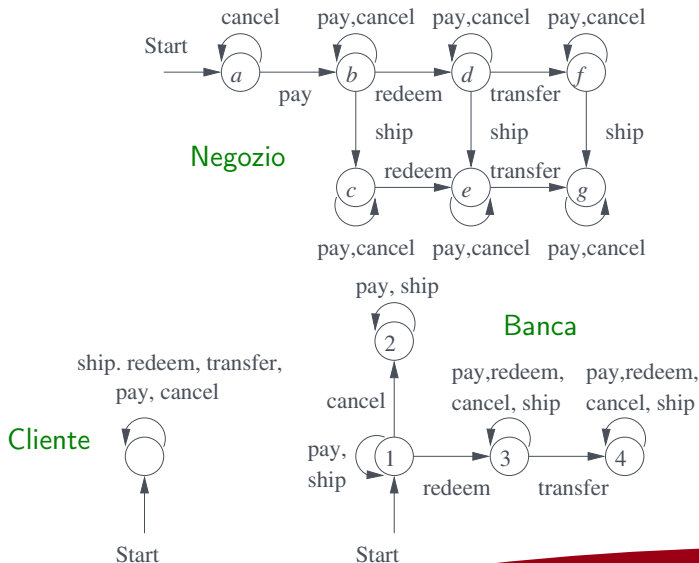
- Il **cliente** **paga** il negozio con moneta elettronica
- Il **cliente** può **cancellare** la moneta elettronica
- Il **negozio** riceve il pagamento e **spedisce** il prodotto al cliente
- Per completare il pagamento, il **negozio** **riscatta** la moneta elettronica
- La **banca** controlla la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

Modelliamo l'esempio

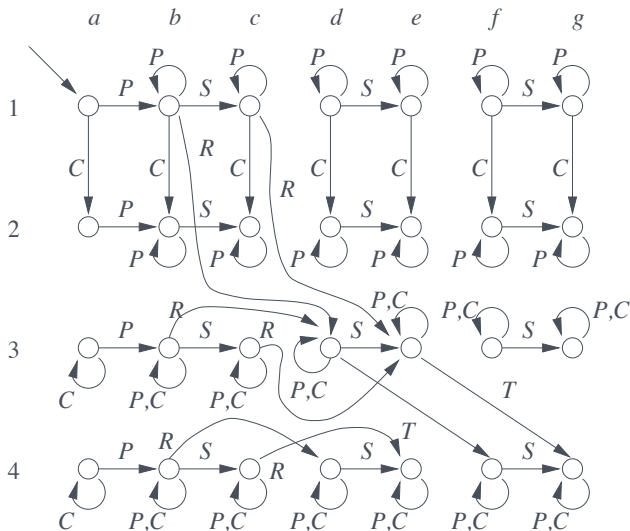


- Ogni automa reagisce solo ad alcuni messaggi:
 - Il cliente può ignorare riscatta e trasferisci
 - La banca può ignorare paga e spedisci
- Dobbiamo gestire anche comportamenti inattesi:
 - Cosa facciamo se cliente paga due volte?
- La definizione formale di automa prescrive che si debba reagire ad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

Completiamo gli automi



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

- **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 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

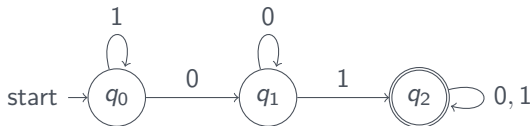
$$A = (Q, \Sigma, \delta, q_0, F)$$

- Q è un insieme finito di **stati**
- Σ è un **alfabeto finito** (= simboli in input)
- δ 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

Modellare il comportamento di un distributore di bibite con un DFA. Il modello deve rispettare le seguenti specifiche:



- Costo della bibita: 40 centesimi
- Monete utilizzabili: 10 centesimi, 20 centesimi
- Appena le monete inserite raggiungono o superano il costo della bibita, il distributore emette una lattina
- Il distributore dà il resto (se serve) subito dopo aver emesso la lattina