Automi e dei linguaggi formali

Testi di riferimento

- **Libro di testo principale**: *Automi, linguaggi e calcolabilità*, J. E. Hopcroft, R. Motwani, and J. D. Ullman, Addison-Wesley, 2003.
- **Altro libro**: *Compilers: principles, techniques, and tools*, A.H. Aho, M. S. Lam, R. Sethi, J. D. Ullman, Pearson/Addison-Wesley, 2007 (seconda edizione).
- I lucidi su questa parte del corso sono basati su quelli della Prof. Francesca Rossi (basati a loro volta su quelli di Gösta Grahne e David Ford).

Qual è il pattern?

Un robot lancia in aria una moneta e occorre indovinare se viene testa o croce. Sembra esserci tuttavia una sequenza, un pattern, ripetuta nel modo in cui la moneta appare. È un gioco truccato? Dati i seguenti risultati relativi ai tiri della moneta (t=testa, c=croce):

ttcttctttccttttcctccctttttcttt
ccctttccctttttcctcccccctcctccc
tttcctttctttttttttccttttccccttt
ttcccccccc

qual è il pattern per prevedere i risultati dei prossimi tiri?

Qual è il pattern?

ttc/ttc/ttt/cct/ttt/cct/ccc/ttt/ttc/ttt
ccc/ttt/ccc/ttt/ttt/cct/ccc/cct/cct/ccc
ttt/cct/ttc/ttt/ttt/ttt/cct/ttc/ccc/ttt
ttc/ccc/ccc

I primi due tiri ogni tre danno sempre lo stesso risultato

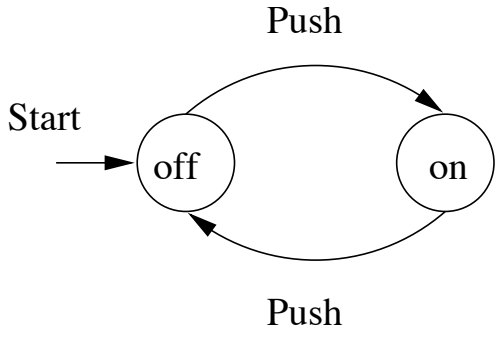
Automi a stati finiti

Gli automi a stati finiti sono usati come modello per

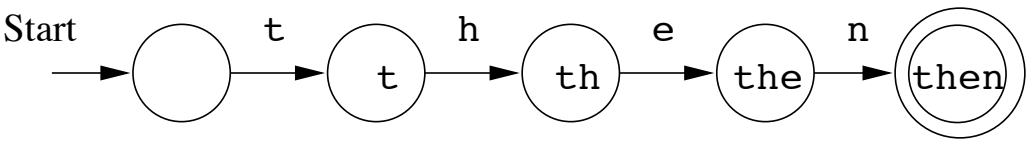
- Software per la progettazione di circuiti digitali.
- Analizzatori lessicali di un compilatore.
- Ricerca di parole chiave in un file o sul web.
- Software per verificare sistemi a stati finiti, come protocolli di comunicazione.

Esempi

- Esempio: automa a stati finiti per un interruttore on/off



- Esempio: automa a stati finiti che riconosce la stringa `then`



Rappresentazioni strutturali

Ci sono vari modi di specificare una macchina

- **Grammatiche**:
  Una regola come $E \Rightarrow E + E$ specifica un'espressione aritmetica
  $Coda \Rightarrow Persona.Coda$
  dice che una coda è costituita da una persona seguita da una coda.
- **Espressioni regolari**:
  Denotano la struttura dei dati, per esempio:
  `'[A-Z][a-z]*[][A-Z][A-Z]'`
  è compatibile con (matches) `Ithaca NY`
  non è compatibile con `Palo Alto CA`
- Domanda: Quale espressione è compatibile con
  `Palo Alto CA`?

Concetti di base

- **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:** Sequenza finita di simboli da un alfabeto Σ, per es. 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 posizioni per i simboli nella stringa.
$|w|$ denota la lunghezza della stringa w
$|0110| = 4, |\epsilon| = 0$
Potenze di un alfabeto: Σ^k = insieme delle stringhe di lunghezza k con simboli da Σ
Esempio: $\Sigma = \{0, 1\}$
$\Sigma^1 = \{0, 1\}$
$\Sigma^2 = \{00, 01, 10, 11\}$
$\Sigma^0 = \{\epsilon\}$
Domanda: Quante stringhe ci sono in Σ^3?

L'insieme di tutte le stringhe su Σ è denotato da Σ^*
$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$
Anche:
$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots$
$\Sigma^* = \Sigma^+ \cup \{\epsilon\}$
Se Σ è finito, Σ^* è *numerabile*, ovvero esiste una corrispondenza biunivoca tra Σ^* e N
Concatenazione: Se x e y sono stringhe, allora xy è la stringa ottenuta rimpiazzando una copia di y immediatamente dopo una copia di x
$x = a_1 a_2 \ldots a_i, y = b_1 b_2 \ldots b_j$
$xy = a_1 a_2 \ldots a_i b_1 b_2 \ldots b_j$
Esempio: $x = 01101, y = 110, xy = 01101110$
Nota: Per ogni stringa x

$$x\epsilon = \epsilon x = x$$

Linguaggi

Definizione: Se Σ è un alfabeto, e $L \subseteq \Sigma^*$, allora L è un linguaggio
Esempi di linguaggi:

- L'insieme delle parole italiane legali
- L'insieme dei programmi C legali
- L'insieme delle stringhe che consistono di n zeri seguiti da n uni

$$\{\epsilon, 01, 0011, 000111, \ldots\}$$

Altri esempi

- L'insieme delle stringhe con un numero uguale di zeri e di uni

$$\{\epsilon, 01, 10, 0011, 0101, 1001, \ldots\}$$

- L_P = insieme dei numeri binari il cui valore è primo

$$\{10, 11, 101, 111, 1011, \ldots\}$$

- Il linguaggio vuoto $\emptyset$
- Il linguaggio $\{\epsilon\}$ consiste della stringa vuota

Nota: $\emptyset \neq \{\epsilon\}$
Nota: L'alfabeto Σ è sempre finito

Problemi

- La stringa w è un elemento di un linguaggio L?
- Esempio: Dato un numero binario, è primo = è un elemento di L_P?
- È $11101 \in L_P$? Che risorse computazionali sono necessarie per rispondere a questa domanda?
- Di solito non pensiamo ai problemi come delle decisioni sì/no, ma come qualcosa che trasforma un input in un output.
- Esempio: Fare il parsing di un programma C = controllare se il programma è corretto, e se lo è, produrre un albero di parsing.

Automi a stati finiti deterministici

Un DFA è una quintupla

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

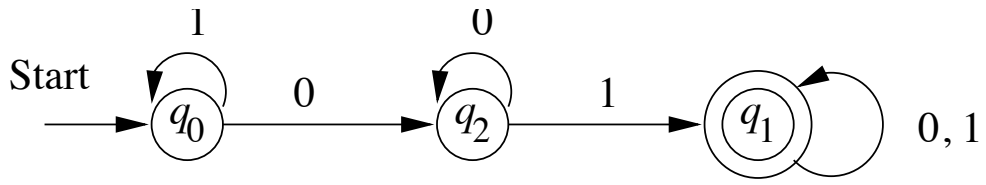
- Q è un insieme finito di *stati*
- Σ è un *alfabeto finito* (= simboli in input)
- δ è una *funzione di transizione* $(q, a) \mapsto p$
- $q_0 \in Q$ è lo *stato iniziale*
- $F \subseteq Q$ è un insieme di *stati finali*

Esempio

Esempio: Un automa A che accetta
$L = \{x01y : x, y \in \{0,1\}^*\}$
L'automa $A = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$ come una *tabella di transizione*:

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

L'automa come un *diagramma di transizione*:

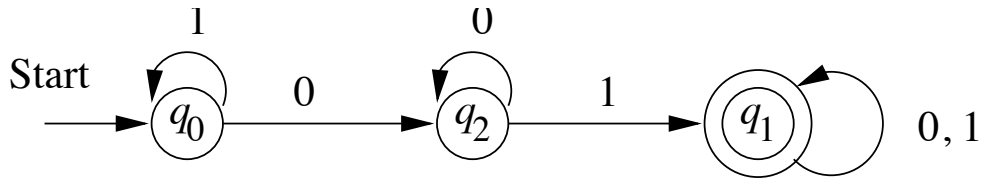


Accettazione

Un automa a stati finiti (FA) *accetta* una stringa $w = a_1 a_2 \cdots a_n$ se esiste un cammino nel diagramma di transizione che

1. Inizia nello stato iniziale
2. Finisce in uno stato finale (di accettazione)
3. Ha una sequenza di etichette $a_1 a_2 \cdots a_n$

Esempio: L'automa a stati finiti



accetta ad esempio la stringa 01101