

Macchina di Turing

Dipartimento di Elettronica e Informazione
Politecnico di Milano

17 marzo 2017

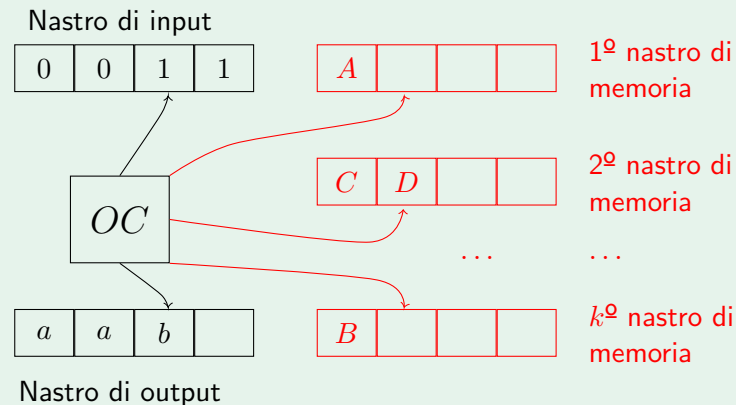
Un modello di calcolo universale

Macchina di Turing

- Gli AP sono più potenti degli FSA (= maggiori capacità riconoscitive), ma mostrano ancora limitazioni
- Esaminiamo un modello di calcolo con maggiori capacità
 - Saremo in grado di effettuare “qualunque” calcolo
 - Esamineremo qual é il costo di questa generalità espressiva
- Il modello è la *Macchina di Turing* (MT) (Alan Turing 1912-1954)
 - Uno dei primi modelli di calcolo: sorprendentemente “semplice” nella sua efficacia
 - Esaminiamo ora il funzionamento come riconoscitore e traduttore, in seguito le proprietà universali del calcolo automatico
- Esaminiamo prima la MT a k nastri per semplicità

Macchina di Turing

Modello a k -nastri



Descrizione informale

Componenti

- Insieme di stati dell'OC, alfabeto I e O come per gli AP
- Ogni nastro di memoria può avere un suo alfabeto dedicato o meno: considereremo un unico alfabeto Γ
- Per convenzione storica e semplicità di formalizzazione, i nastri sono sequenze infinite di celle
 - Solo una quantità finita è inizializzata con un valore sensato
 - Celle restanti contenenti uno spazio vuoto o *blank*: $\bar{}$
- Le tutte testine possono scorrere sui nastri o rimanere ferme
- Configurazione di MT: stato dell'OC e contenuto dei nastri

Transizione della MT

In funzione di ...

- Lettura del carattere sotto la testina del nastro di input
- Lettura dei caratteri sotto le testine del nastro di memoria
- Stato dell'OC

... azione conseguente

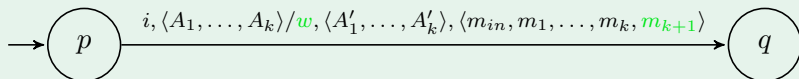
- Cambiamento di stato dell'OC
- Scrittura di un carattere su ogni nastro di memoria
- **Scrittura di un carattere sul nastro di uscita**
- Spostamento delle testine
 - Testine di memoria e testina di ingresso: 1 posizione a sinistra (L), destra (R) o stanno ferme (S)
 - Testina di output: convenzionalmente solo S o R, se si sposta scrive sempre una lettera o un \bar{x}

Transizione della MT

Riconoscitore e traduttore

- Transizione $\delta : \mathbf{Q} \times (\mathbf{I} \cup \{\mathfrak{b}\}) \times \Gamma^k \rightarrow \mathbf{Q} \times \Gamma^k \times \{\mathbf{L}, \mathbf{S}, \mathbf{R}\}^{k+1}$
- Traduzione $\eta : \mathbf{Q} \times (\mathbf{I} \cup \{\mathfrak{b}\}) \times \Gamma^k \rightarrow (\mathbf{O} \cup \{\mathfrak{b}\})$
 - Per quale ragione \mathbf{O} non è meno generale di \mathbf{O}^* ?

Convenzione grafica



- Mossa $m_{in} \in \{\mathbf{L}, \mathbf{S}, \mathbf{R}\}$: testina di input
- Nastri da $i = 1$ a $i = k \rightarrow$ memoria, mosse $m_i \in \{\mathbf{L}, \mathbf{S}, \mathbf{R}\}$
- Nastro $k + 1 \rightarrow$ output, $m_{k+2} \in \{\mathbf{S}, \mathbf{R}\}$

Transizione della MT

Configurazione iniziale

- Tutti i nastri di memoria pieni di \bar{b} , tranne nella posizione iniziale della rispettiva testina dove c'è Z_0
- Per farvi riferimento, la posizione delle testine conta da 0
- Stato iniziale dell'organo di controllo q_0
- Stringa in ingresso x scritta a partire dalla 0^a cella del nastro di ingresso, seguita e preceduta da \bar{b}
- **Nastro di output riempito con \bar{b}**

Transizione della MT

Configurazione finale e accettazione

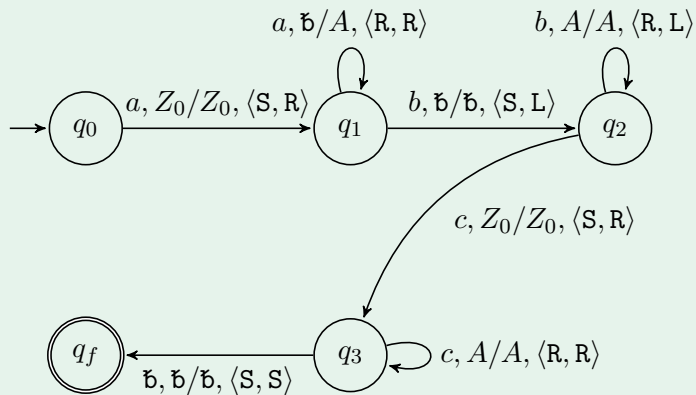
- Stati di accettazione come per FSA, AP: $\mathbf{F} \subseteq \mathbf{Q}$
- Per comodità (convenzione) la δ, η non è definita a partire dagli stati finali : $\forall q \in \mathbf{F}, \delta(q, \dots) = \perp, \eta(q, \dots) = \perp$
- La MT si ferma in uno stato q se $\delta(q, \dots) = \perp$
- La stringa x in ingresso è accettata se e solo se:
 - In numero finito di transizioni, la macchina si ferma in $q \in \mathbf{F}$

Mancata accettazione

- Una stringa x in ingresso non è accettata se
 - la macchina si ferma in uno stato $\notin \mathbf{F}$
 - *la macchina non si ferma*
 - "Somiglia" agli AP, ma ... esiste una MT loop-free?

MT: esempi

Riconoscere $L = \{a^n b^n c^n, n > 0\}$



Esempio di traduzione: Incremento di un numero decimale

Schema di funzionamento

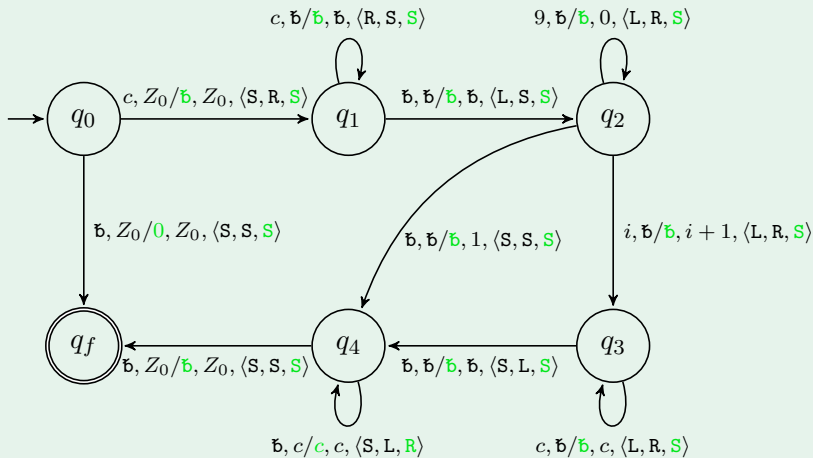
- 1 Scorri l'input fino alla fine
- 2 Scorrendo all'indietro l'input, scrivi 0 sul nastro di memoria fino alla prima cifra diversa da 9
- 3 Scrivi il successore della cifra corrente sul nastro di memoria, dopodichè copia le cifre rimanenti dall'ingresso su di esso
- 4 Scorrendo al contrario il nastro di memoria, copia le cifre in uscita

Notazione sintetica

- 1 i : una qualunque cifra tra 0 e 8
- 2 c : una qualunque cifra decimale
- 3 $i+1$: il successore della cifra decimale i

MT: esempi

Incrementatore decimale (con $\tau(\varepsilon) = 0$)



Proprietà di chiusura

Valide

- \cap : **OK**, una MT che simula l'esecuzione di due "in serie"
- \cup : **OK**, una MT che simula l'esecuzione di due "in parallelo"
- \cdot (concatenazione): **OK**, simile a \cap
- $*$: **OK** vedi concatenazione

Non valide

- Complemento: **NO**. (Dimostrazione nelle prossime lezioni)
 - Se potessi avere MT loop-free sarebbe facile: individuo l'insieme degli stati di arresto, lo partiziono in finali e non finali, scambio gli insiemi ed è fatta
 - Il problema è nelle *computazioni che non terminano*

Modelli equivalenti alla MT

Turing completezza

- È possibile costruire modelli di calcolo equivalenti alla MT
 - Una MT con nastro bidimensionale (mosse {N,S,O,E})
 - Una MT con k testine per nastro
 - Un AP con due pile
- Idea generale: per mostrare che sono equivalenti si mettono in corrispondenza biunivoca le transizioni di una MT “classica” a quelle del modello di calcolo proposto
- Se un modello di calcolo è in grado di simulare una MT, viene detto *Turing completo*
 - La semantica di quasi tutti i linguaggi di programmazione ^a
 - Il meccanismo di risoluzione dei page fault dell'MMU x86_64^a

^aassumendo risorse di calcolo infinite

MT a nastro singolo

Nastro singolo \neq 1 nastro di memoria

- Una particolare variante della MT è quella a nastro singolo: input, output e memoria sono tutti sullo stesso nastro
 - È il modello inizialmente ideato da Turing
 - La posizione delle testine è tenuta usando un carattere speciale sul nastro, e memorizzando il carattere sotto la testina nell'OC
 - Il contenuto dei diversi nastri è delimitato da altri caratteri speciali, si sequenzializzano le operazioni su nastri diversi

MT e modelli di calcolo classico

Differenze con modelli più “realistici”

- Una MT è in grado di simulare il comportamento di una macchina di von Neumann (anch'essa “astratta”)
- La differenza principale è la modalità di accesso alla memoria
 - Una macchina di von Neumann accede per indirizzo ai dati, mentre una MT scorre il nastro opportuno
 - Nessun cambio a livello di capacità computazionali (= problemi risolvibili)
- Spesso cambiare modello di calcolo ha un significativo impatto in complessità (= tempo/spazio impiegati) del calcolo