

# Fondamenti di Informatica

Prof. V.L. Plantamura  
Informatica e Comunicazione Digitale  
a.a. 2006-2007

---

---

---

---

---

---

---

---

## Ipotesi fondamentale

- Tesi di Church  
Qualunque algoritmo può essere espresso sotto forma di matrice funzionale ed eseguito dalla corrispondente Macchina di Turing

---

---

---

---

---

---

---

---

## Irrisolubilità

- Un problema è non risolubile algoritmicamente se nessuna Macchina di Turing è in grado di fornire la soluzione al problema in tempo finito

---

---

---

---

---

---

---

---

## Macchina applicabile

- Diciamo che la macchina è *applicabile* all'informazione iniziale A e che essa l'ha trasformata nell'informazione risultante B se, dopo un numero finito di cicli, la macchina si ferma, avendo ricevuto un ordine di Stop, e sul nastro si ha la rappresentazione dell'informazione finale B.

---

---

---

---

---

---

---

---

## Macchina non applicabile

- Diciamo che la macchina è *non applicabile* all'informazione iniziale A se non riceve mai un ordine di stop e quindi non si ferma mai.

---

---

---

---

---

---

---

---

## Macchina di Turing

- Se l'esecutore conosce come funziona la macchina di Turing è sufficiente darle l'informazione iniziale e la matrice funzionale della macchina affinché egli possa eseguire l'algoritmo che essa descrive.

---

---

---

---

---

---

---

---

## Algoritmo di imitazione

- Il processo di imitazione di una macchina in base alla sua matrice può essere eseguito anche da chi non conosca assolutamente le macchine di Turing.

---

---

---

---

---

---

---

---

## Algoritmo di imitazione

1. Esaminare la cella del nastro immediatamente al disotto della quale figura un simbolo
2. Trovare, nella matrice funzionale, la colonna contrassegnata col simbolo che appare sotto la cella esaminata
3. Trovare, nella matrice funzionale, la riga contrassegnata col simbolo che appare nella cella esaminata e annotare la tripla di simboli che compare all'incrocio di questa riga e della colonna individuata con l'istruzione precedente

---

---

---

---

---

---

---

---

## Algoritmo di imitazione

4. Rimpiazzare la lettera nella cella esaminata col primo simbolo della tripla
5. Se il secondo simbolo della tripla è !, fermarsi
6. Se il secondo simbolo della tripla è F, rimpiazzare il simbolo sotto la cella esaminata col terzo simbolo della tripla

---

---

---

---

---

---

---

---

## Algoritmo di imitazione

- Se il secondo simbolo della tripla è S, cancellare il simbolo sotto la cella esaminata e imprimere il terzo simbolo della tripla sotto la prima cella immediatamente a sinistra
- Se il secondo simbolo della tripla è D, cancellare il simbolo sotto la cella esaminata e imprimere il terzo simbolo della tripla sotto la prima cella immediatamente a destra
- Passare all'istruzione 1

---

---

---

---

---

---

---

---

## Macchina di Turing Universale

- La macchina di Turing Universale è una macchina capace di eseguire il lavoro di ogni altra macchina di Turing

---

---

---

---

---

---

---

---

## Difficoltà n.1

- Non possiamo porre direttamente la matrice funzionale e la configurazione iniziale della macchina imitata sul nastro della macchina universale, perché anche in questa, come in tutte le macchine di Turing, l'informazione è disposta sul nastro unidimensionalmente mentre le matrici sono bidimensionali e nelle configurazioni il simbolo di stato è posto al di sotto del nastro.

---

---

---

---

---

---

---

---

## Difficoltà n. 2

- La macchina universale impiega un alfabeto esterno finito. Con questo alfabeto noi dobbiamo poter rappresentare tutte le possibili matrici e configurazioni.

---

---

---

---

---

---

---

---

## Sequenza funzionale

- Invece di una matrice bidimensionale con  $k$  righe ed  $m$  colonne, utilizziamo una sequenza di  $m \cdot k$  quintuple consecutive di simboli

$$(S, Q, s, P, q)$$

---

---

---

---

---

---

---

---

## Codifica dei simboli

- Deve esistere un algoritmo in base al quale distinguere fra loro i gruppi codificati per lo scorrimento  $P$ , per i simboli di stato  $Q$  e per i simboli dell'alfabeto esterno  $S$ .

---

---

---

---

---

---

---

---

## Codifica

- S = 101
- F = 1001
- D = 10001
- Alfabeto esterno = 100...001 con un numero pari di zeri, maggiore di 2
- Simboli di stato = 100...01 con un numero dispari di zeri, maggiore di 3
- 100001100000110000110001100000001100001...

---

---

---

---

---

---

---

---

## Perché le Macchine di Turing

- Questo modello definisce chiaramente e non ambigualmente il concetto di passo della computazione e quindi di *tempo* richiesto.
- Offre la definizione di unità di memoria (la cella del nastro) e quindi di *spazio* utilizzato.

---

---

---

---

---

---

---

---

## Gioco di Turing

- Si gioca con tre persone, un uomo (A), una donna (B), ed un intervistatore (C) che può essere di entrambi i sessi. L'intervistatore sta in una stanza separato dagli altri - collegato con loro tramite una telescrivente. L'obiettivo del gioco per l'intervistatore e quello di determinare quale degli altri due sia l'uomo e quale la donna.
- Egli li conosce con le etichette X e Y e alla fine del gioco dovrà dire "X è A ed Y è B" oppure "X è B ed Y è A".

---

---

---

---

---

---

---

---

## Gioco di Turing

- All'intervistatore è permesso di porre delle domande ad A e B, ad esempio: "Mi vuol dire per favore X la lunghezza dei suoi capelli?"
- Ora supponiamo che X sia veramente A. L'obiettivo di A nel gioco è quello di cercare di fare in modo che C sbagli identificazione. L'obiettivo del gioco per il terzo giocatore B è quello di aiutare l'intervistatore.
- Ora poniamo la domanda: Cosa accadrà quando una macchina prenderà il posto di A in questo gioco?

---

---

---

---

---

---

---

---

## Gioco di Turing

- Queste domande sostituiscono la nostra originale:
  - **"È una macchina in grado di pensare?"**
  - (Turing 1950)
- Se una "macchina" riesce ad avere un punteggio pari a quello di A o di B, allora si può dire che essa "pensa"?

---

---

---

---

---

---

---

---

## Random Access Machine (RAM)

- **Un nastro di input**
  - di lunghezza infinita, suddiviso in celle; su tale nastro è consentita la sola operazione di lettura, in modo sequenziale
- **Un nastro di output**
  - di lunghezza infinita, suddiviso in celle; su tale nastro è consentita la sola operazione di scrittura, in modo sequenziale
- Un **programma** che viene eseguito sequenzialmente
- Una **memoria** di dimensione infinita su cui conservare i risultati intermedi

---

---

---

---

---

---

---

---

## Random Access Machine (RAM)

Si assume che:

1. Il programma non possa modificare se stesso
2. Tutti i calcoli avvengono utilizzando una locazione fissa di memoria detta accumulatore
3. Ogni locazione di memoria ed ogni cella del nastro di input ed output siano in grado di contenere un simbolo arbitrario
4. Il programma possa accedere alle singole locazioni di memoria in ordine arbitrario

---

---

---

---

---

---

---

---

## Accesso Diretto

- Il tempo di accesso ad una cella di memoria deve essere indipendente dalla cella stessa

---

---

---

---

---

---

---

---

## Complessità computazionale

- Esistono due criteri per determinare la quantità di tempo e di spazio richieste durante l'esecuzione di un programma RAM:
  - Il criterio di costo uniforme
  - Il criterio di costo logaritmico

---

---

---

---

---

---

---

---



## Costo uniforme

- L'esecuzione di ogni istruzione del programma richiede una quantità di tempo costante
- Lo spazio richiesto per l'utilizzo di un registro di memoria è di una unità, indipendentemente dal suo contenuto

---

---

---

---

---

---

---

---

## Costo Logaritmico

- Attribuiamo ad ogni istruzione un costo di esecuzione che dipende dalla dimensione degli operandi.
- Tale criterio è così chiamato perché per rappresentare un numero intero  $n$  occorrono  $\lfloor \log n \rfloor + 1$  bit.

---

---

---

---

---

---

---

---