
Università Roma Tre – Dipartimento di Matematica e Fisica – Corso di Laurea in Matematica

Appunti del corso di Informatica 1 (IN110 – Fondamenti)

3 – Modelli di calcolo

Marco Liverani
(liverani@mat.uniroma3.it)

Sommario

- Modelli di calcolo
- Cenni sulla macchina di Turing
- Un esempio di grafo degli stati e di matrice di transizione
- Cenni sulla macchina di Von Neumann

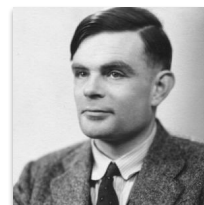
Modelli di calcolo

- Prima di specializzarci nella costruzione di algoritmi (e poi di programmi) per un particolare calcolatore è bene convincersi che **non** esiste un unico *modello di calcolo*
- Ogni modello richiede un differente approccio alla soluzione dei problemi, proponendo un **modello teorico di macchina di calcolo** (di esecutore) con *caratteristiche e capacità* differenti
- Basarsi su un modello di macchina astratta è necessario per poter **confrontare l'efficienza** di due differenti *algoritmi*, a prescindere dalla loro implementazione e dalla velocità del computer su cui eseguiremo il programma codificato in linguaggio macchina
- Basandoci su un modello di calcolo **astratto**, ma **semplificato**, è più semplice dimostrare in termini rigorosi la *correttezza* di un determinato algoritmo

La macchina di Turing

1

- Ideata nel 1936 dal matematico inglese **Alan Turing** (1912–1954) una delle figure più importanti che hanno contribuito alla definizione e allo sviluppo dell'informatica
- È una macchina **astratta** basata su due componenti:
 - Un **nastro infinito** (da questa caratteristica ne segue il fatto che la macchina di Turing non è realizzabile praticamente) su cui la macchina può leggere e scrivere mediante
 - Una **testina di lettura e scrittura** che, scorrendo sul nastro, è in grado di leggerne e modificarne il contenuto
- La macchina di Turing è un *Automa a Stati Finiti Deterministico*



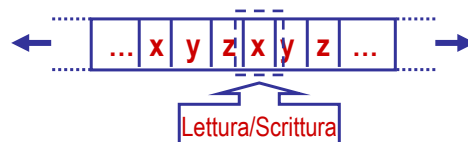
La macchina di Turing

2

- Il funzionamento della macchina è basato sul **cambiamento di stato** sulla base del contenuto (del simbolo) che è presente sul nastro in corrispondenza della testina di lettura/scrittura
- Dopo aver letto il contenuto della posizione corrente del nastro, la macchina, sulla base dello stato in cui si trova, è in grado di passare in un altro stato, scrivere qualcosa nella posizione corrente del nastro ed infine spostarsi a destra o a sinistra sul nastro stesso
- Il **nastro è infinito**, mentre gli **stati sono in quantità finita** (è un automa **a stati finiti** deterministico)
- Il fatto che sia **deterministico** sta ad indicare che ad ogni coppia “stato/simbolo” che viene incontrata dalla macchina, è **univocamente** determinata l’azione e lo stato compiuto dalla macchina stessa; in questo modo la macchina può eseguire solo un’operazione per volta

La macchina di Turing

3



- Per ogni problema che si intende risolvere è necessario progettare una macchina di Turing adeguata
- Per farlo è necessario definire:
 - L'**alfabeto** dei simboli che è possibile leggere e scrivere sul nastro infinito
 - Gli **stati** in cui si può trovare la macchina
 - Le **transizioni** da uno stato ad un altro
 - Lo **stato iniziale** ed un insieme di **stati finali**
- Per rappresentare l’automata possiamo disegnare un **grafo degli stati**, ovvero una **matrice di transizione**

La macchina di Turing

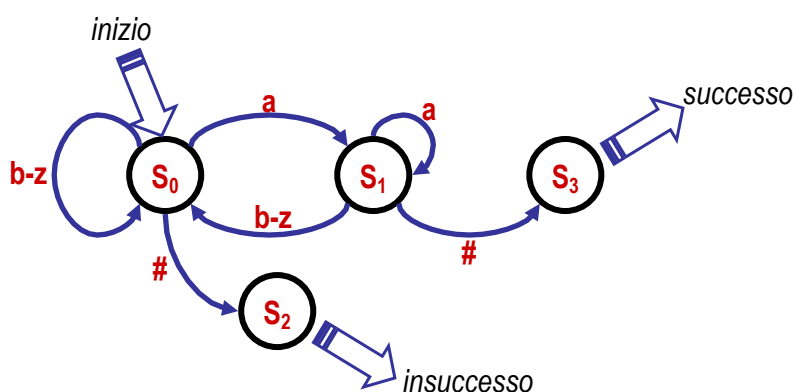
4

- **Esempio:** *letta una stringa di caratteri alfabetici, stabilire se termina con la lettera "a"*
- **Alfabeto:** $a, b, c, \dots, x, y, z, \#$ (il carattere " $\#$ " marcherà la fine della stringa)
- **Strategia risolutiva:** partendo dal primo carattere della stringa scorro verso destra il nastro fino a quando non incontro il carattere " $\#$ "
 Ogni volta che trovo un carattere " a " la macchina si pone in uno *stato di preallarme* (S_1) e torna nello *stato di "quiete"* (S_0) quando incontra una lettera dalla " b " alla " z "
 Se quando incontra il " $\#$ " si trova nello stato S_0 allora la stringa non termina con la lettera " a " (passo in S_2), altrimenti, se si trova in S_1 , vuol dire che la stringa termina con la lettera " a " (passo in S_3)

La macchina di Turing

5

- **Grafo degli stati:**



La macchina di Turing

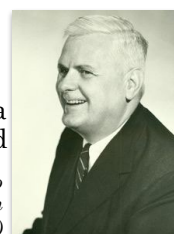
6

• **Matrice di transizione:**

	S₀	S₁	S₂	S₃
a	-, Dx, S₁	-, Dx, S₁	n.d.	n.d.
b-z	-, Dx, S₀	-, Dx, S₀	n.d.	n.d.
#	-, -, S₂	-, -, S₃	No	Sì

• **Tesi di Church-Turing:**

- Anche assumendo altri modelli di calcolo, diversi dalla macchina di Turing, si afferma che tutti questi modelli sono fra loro **equivalenti**
- Un problema **calcolabile** secondo il modello della macchina di Turing è calcolabile anche rispetto ad altri modelli

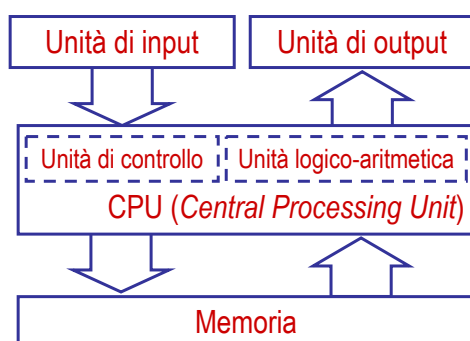


*Il logico-matematico
Alonzo Church
(1903 - 1995)*

Il modello di macchina di Von Neumann

1

- Gli algoritmi che progetteremo si basano sulle caratteristiche e sulle capacità di base di un calcolatore basato su un *modello di calcolo ideale*, noto come **Macchina di Von Neumann**, proposta nel 1945 dallo scienziato ungherese/statunitense John Von Neumann (1903-1957)



Il modello di macchina di Von Neumann 2

- Nel modello di Von Neumann è facile riconoscere gli elementi costitutivi di ogni moderno calcolatore
- Le componenti sono le seguenti:
 - Le **unità di input** tramite cui la macchina acquisisce informazioni dall'esterno
 - Le **unità di output** tramite cui la macchina produce (stampa) informazioni all'esterno
 - L'**unità centrale di elaborazione** (CPU) che elabora le istruzioni del programma (i passi dell'algoritmo), composta da due elementi:
 - **Unità di controllo**: stabilisce l'ordine con cui devono essere eseguite le operazioni
 - **Unità logico-aritmetica**: esegue operazioni aritmetiche e risolve espressioni logiche
 - La **memoria** in cui l'unità centrale deposita ed estrae le informazioni per poterle elaborare
- I dati e il programma eseguito dalla macchina sono memorizzati nella stessa unità di memoria