



Nel 1936 il matematico inglese Alan Turing propose l'idea di una macchina astratta che fosse capace di eseguire ogni tipo di calcolo su numeri e simboli. La tesi di Church postula che la classe delle funzioni calcolabili, secondo il concetto intuitivo di algoritmo, coincide con la classe delle funzioni Turing-calcolabili cioè calcolabili con una Macchina di Turing.

Una MdT è definita da:

1. un nastro infinito
2. una testina di lettura/scrittura
3. un'unità di controllo
4. Il nastro infinito

Il **nastro infinito** che può essere considerato come il supporto di memorizzazione delle informazioni (memoria esterna) è suddiviso in celle, in una cella può essere contenuto un simbolo preso da un alfabeto opportuno; un alfabeto è semplicemente un insieme di simboli che comprende anche il blank che corrisponde alla cella vuota.

#### La **testina di lettura/scrittura**

La macchina è dotata di una testina di lettura/scrittura in grado di leggere e scrivere il contenuto della cella del nastro su cui si trova;

#### L'**unità di controllo**

La macchina è dotata di un'unità di controllo che evolve attraversando un insieme finito di stati interni, a partire da uno stato iniziale fino a raggiungere uno stato finale. Lo stato interno rappresenta una situazione in cui si trova la macchina durante l'esecuzione e come per uno stato della mente di un essere umano, lo stato interno di una MdT definisce l'ambiente in cui una decisione viene presa.

In sintesi una macchina di Turing (MdT) è definita da un insieme di regole che definiscono il comportamento della macchina su un nastro di input-output (lettura e scrittura). Il nastro può essere immaginato come un nastro di carta di lunghezza infinita, diviso in quadratini dette celle. Ogni cella contiene un simbolo oppure è vuota. Una MdT ha una testina che si sposta lungo il nastro leggendo, scrivendo oppure cancellando simboli nelle celle del nastro.

La macchina analizza il nastro, una cella alla volta, iniziando dalla cella su cui è posizionata la testina. Ad ogni passo, la macchina legge un simbolo sul nastro e a partire dal suo stato interno corrente:

1. determina il suo prossimo stato interno
2. scrive un simbolo sul nastro
3. decide se spostare o meno la testina a sinistra o a destra di una posizione

Il comportamento di una MdT può essere programmato definendo un insieme di regole, o quintuple, del tipo: (stato-interno-corrente, simbolo-letto, prossimo-stato-interno, simbolo-scritto, direzione testina).

Ciascuna quintupla associa ad ogni coppia statoInternoCorrente, simboloLetto

una terna: prossimoStatoInterno, simboloScritto, direzioneTestina.

In sintesi una MdT effettua i suoi calcoli in questo modo:

Inizialmente il nastro contiene una sequenza finita di simboli, detta sequenza di ingresso. La MdT è nel suo stato interno iniziale con la testina posizionata su una cella del nastro contenente un certo simbolo. A

partire da questa configurazione iniziale, la MdT effettua una serie di azioni (chiamate mosse) seguendo rigorosamente il suo insieme di regole. Se la macchina raggiunge uno stato interno per cui non esiste nessuna quintupla per la coppia: stato-interno-corrente, simbolo-letto allora la MdT si ferma e termina la sua computazione.

**Esempio: cambiamo A in B**

Vogliamo programmare una Macchina di Turing che, dato sul nastro di input una stringa di A e B, sostituisce ogni occorrenza di A con B e viceversa. Assumendo che la testina sia posizionata sul primo simbolo della stringa dobbiamo cambiare una A in B (o viceversa), spostare la testina sul prossimo carattere, cambiare A con B (o viceversa) ... continua

Per esempio la quintupla (0, A, 1, B, S) indica che se la macchina si trova nello stato interno 0 e legge sul nastro il simbolo A, allora passa nello stato interno 1, scrive B sul nastro e si sposta a sx.

Una tale MdT può essere definita dal seguente insieme di regole:

Stato interno	Legge	Prossimo stato interno	Scrive	Sposta
0	A	1	B	s
0	B	1	A	s
1	A	0	B	s
1	B	0	A	s

Utilizziamo il seguente simulatore: <http://kangourou.di.unimi.it/2012/turing/myturing.html>

**Programma:**

(0,A) > (1,B,s)

(1,A) > (0,B,s)

(0,B) > (1,A,s)

(1,B) > (0,A,s)

Nastro Iniziale: ABAAABBA

Utilizziamo il seguente simulatore: <http://morphett.info/turing/turing.html#LoadMenu>

**Programma:**

0 A B r 1

0 B A r 1

1 A B r 0

1 B A r 0

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3<sup>a</sup> sez.B spec. Informatica e telecomunicazioni

Data:09/02/2020

Numero progressivo dell'esercizio: es.1

Versione: 1.0

Programmatore/i: Antonio Pio Ferrovia

Sistema Operativo: Windows 10

Compilatore/Interprete: <http://morphett.info/turing/turing.html#LoadMenu>

Obiettivo didattico: programmare la macchina di Turing

Obiettivo del programma: Controllare se una stringa binaria è palindroma (prg presente negli esempi del sito <http://morphett.info/turing/turing.html#LoadMenu>)

<b>0 0 _ r 1o</b>	Se la MdT è nello stato "0" legge 0 cambia stato in "1o" ( cancella scrivendo "spazio")
<b>0 1 _ r 1i</b>	Se la MdT è nello stato "0" legge 1 cambia stato in "1i" ( cancella scrivendo "spazio")
<b>0 _ _ * accept</b>	Se la MdT è nello stato "0" legge "spazio" passa allo stato accept(fine del programma con esito positivo)
<b>1o _ _ l 2o</b>	Se la MdT è nello stato "1o" legge "spazio" torna indietro
<b>1o * * r 1o</b>	Se la MdT è nello stato "1o" legge "qualsiasi valore" passa avanti
<b>1i _ _ l 2i</b>	Se la MdT è nello stato "1i" legge "spazio" e torna indietro
<b>1i * * r 1i</b>	Se la MdT è nello stato "1i" legge "qualsiasi valore" passa avanti
<b>2o 0 _ l 3</b>	Se la MdT è nello stato "2o" legge 0 cambia stato in "3" ( cancella scrivendo "spazio")
<b>2o _ _ * accept</b>	Se la MdT è nello stato "2o" legge "spazio" passa allo stato accept(fine del programma con esito positivo)
<b>2o * * * reject</b>	Se la MdT è nello stato "2o" legge "qualsiasi valore" che è diverso da 0 cambia stato in "reject"(fine del programma con esito negativo)
<b>2i 1 _ l 3</b>	Se la MdT è nello stato "2i" legge 1 cambia stato in "3" ( cancella scrivendo "spazio")
<b>2i _ _ * accept</b>	Se la MdT è nello stato "2i" legge "spazio" passa allo stato accept(fine del programma con esito positivo)
<b>2i * * * reject</b>	Se la MdT è nello stato "2i" legge "qualsiasi valore" che è diverso da 0 cambia stato in "reject"(fine del programma con esito negativo)
<b>3 _ _ * accept</b>	Se la MdT è nello stato "3" legge "spazio" passa allo stato accept(fine del programma con esito positivo)
<b>3 * * l 4</b>	Se la MdT è nello stato "3" legge "qualsiasi valore" torna indietro, andando nello stato "4"
<b>4 * * l 4</b>	Se la MdT è nello stato "4" legge "qualsiasi valore" torna indietro
<b>4 _ _ r 0</b>	Se la MdT è nello stato "4" legge "spazio" passa avanti, tornando allo stato "0"
<b>accept * : r accept2</b>	Se la MdT è nello stato "accept" legge "qualsiasi valore" passa avanti,aggiungendo il carattere "."
<b>accept2 * ) * halt-accept</b>	Se la MdT è nello stato "accept2" legge "qualsiasi valore" si ferma,aggiungendo il carattere ")"(concludendo il programma)
<b>reject _ : r reject2</b>	Se la MdT è nello stato "reject" legge "spazio" passa avanti,aggiungendo il carattere "."
<b>reject * _ l reject</b>	Se la MdT è nello stato "reject" legge "qualsiasi valore" torna indietro
<b>reject2 * ( * halt-reject</b>	Se la MdT è nello stato "reject2" legge "qualsiasi valore" si ferma,aggiungendo il carattere "("(concludendo il programma)

**ITIS-LS "Francesco Giordani" Caserta**

**Anno scolastico:** 2019/2020

**Classe 3<sup>^</sup> sez.B spec. Informatica e telecomunicazioni**

**Data:**09/02/2020

**Numero progressivo dell'esercizio:** es.2

**Versione:** 1.0

**Programmatore/i:** Antonio Pio Ferrovia

**Sistema Operativo:** Windows 10

**Compilatore/Interprete:** <http://morphett.info/turing/turing.html#LoadMenu>

**Obiettivo didattico:** Programmare la macchina di Turing

**Obiettivo del programma:** Data una stringa di 0 e 1 cancellare la stringa

0 \* \* r 0

0 \_ \_ l 1

1 \* \_ l 1

1 \_ \_ \* ac

ac \_ F r ac2

ac2 \_ A r ac3

ac3 \_ T r ac4

ac4 \_ T r ac5

ac5 \_ O r halt-ac

**ITIS-LS "Francesco Giordani" Caserta**

**Anno scolastico:** 2019/2020

**Classe 3<sup>^</sup> sez.B spec. Informatica e telecomunicazioni**

**Data:**09/02/2020

**Numero progressivo dell'esercizio:** es.3

**Versione:** 1.0

**Programmatore/i:** Antonio Pio Ferrovia

**Sistema Operativo:** Windows 10

**Compilatore/Interprete:** <http://morphett.info/turing/turing.html#LoadMenu>

**Obiettivo didattico:** Programmare la macchina di Turing

**Obiettivo del programma:** Dato un numero intero sottrarre 1

```
0 * * r 0
0 _ _ l 1
1 0 9 l i
1 1 0 r a c
1 2 1 r a c
1 3 2 r a c
1 4 3 r a c
1 5 4 r a c
1 6 5 r a c
1 7 6 r a c
1 8 7 r a c
1 9 8 r a c
l i 1 0 r a c
a c * * r a c
a c _ _ r a c 2
a c 2 _ è r a c 3
a c 3 _ _ r a c 4
a c 4 _ i r a c 5
a c 5 _ l r a c 6
a c 6 _ _ r a c 7
a c 7 _ r r a c 8
a c 8 _ i r a c 9
a c 9 _ s r a c 10
a c 10 _ u r a c 11
a c 11 _ l r a c 12
a c 12 _ t r a c 13
a c 13 _ a r a c 14
a c 14 _ t r a c 15
a c 15 _ o r h a l t _ a c
```

### Esercizi di Gara della III edizione

Programmare una macchina di Turing che, dato un nastro iniziale contenente un numero intero  $n$  compreso tra 1 e 9, termina la sua esecuzione lasciando sul nastro  $n$  A consecutive.

Esempi

nastro iniziale	nastro finale
1	A
5	AAAAA
9	AAAAAAAAA

Programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza di  $n$  A consecutive (con  $n > 0$ ), termina la sua esecuzione lasciando sul nastro il numero  $n$ .

Esempi

nastro iniziale	nastro finale
A	1
AAAAAA	6
AAAAAAAAAAAA	11

Programmare una macchina di Turing che, dato un nastro iniziale contenente due numeri interi positivi  $x$  e  $y$  separati da una cella vuota tali che  $x > y$  e  $9 >= y > 0$ , termina la sua esecuzione lasciando sul nastro soltanto la differenza tra  $x$  e  $y$ .

Esempi

nastro iniziale	nastro finale
9-4	5
13-6	7
302-3	299

Indichiamo con  $S$  una sequenza formata da A, B o C ed indichiamo con  $x$  e  $y$  un simbolo che sia A o B.

Programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza del tipo  $xyS$  termina la sua esecuzione lasciando sul nastro la sequenza ottenuta da  $S$  rimpiazzando tutte le occorrenze di  $x$  con  $y$ .

Esempi

nastro iniziale	nastro finale
ABcAB	CBB
BBABC	ABC
BA	

Programmare una macchina di Turing che, dato un nastro iniziale contenente due sequenze di A separate da una D, termina la sua esecuzione lasciando sul nastro la sequenza che contiene il maggior numero di A.

Esempi

nastro iniziale	nastro finale
AADA	AA
AADAAA	AAA
AADAA	DA
AA	A

Indichiamo con S e T due sequenze non vuote e della stessa lunghezza formate da A o B. Programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza del tipo SDT, termina la sua esecuzione lasciando sul nastro la sola sequenza SI se T è un anagramma di S, la sola sequenza NO altrimenti.

Esempi

nastro iniziale	nastro finale
ABADBAA	SI
BABADABBA	SI
ABBDBAA	ABADABB
NO	NO

Programmare una macchina di Turing che, dato un nastro iniziale contenente un numero intero (arbitrariamente grande), termina la sua esecuzione lasciando sul nastro la sola sequenza SI se il numero è divisibile per 3, la sola sequenza NO altrimenti.

Esempi

nastro iniziale	nastro finale
3	SI
27	SI
81	SI
20	NO
7676585	NO

Una sequenza di parentesi si dice bilanciata secondo la seguente definizione induttiva:

la sequenza vuota è bilanciata,

se S e T sono sequenze bilanciate allora anche la sequenza ( S ) T è bilanciata.



Rappresentando ( con B e ) con E, programmare una macchina di Turing che, dato un nastro iniziale contenente una sequenza di B ed E, termina la sua esecuzione lasciando sul nastro la sola sequenza SI se la sequenza è bilanciata, la sola sequenza NO altrimenti.

Esempi

nastro iniziale	nastro finale
BEBE	SI
BBBEEE	SI
BBEBEE	BBEBEBEBEE
SI	SI
BEE	NO
BBEEEB	NO