

Turing, i nastri e le macchine

Giochiamo con i nastri

Avete a disposizione un nastro diviso in sezioni rettangolari, che chiameremo *caselle*, ognuna delle quali può essere vuota oppure contenere la lettera *A* o la lettera *B* (ma non tutte e due contemporaneamente). In particolare il nastro è simile a quello mostrato in Figura 1: contiene una sequenza di *A* seguita da una sequenza di *B*, e voi dovete capire se il numero di *A* scritte sul nastro è uguale al numero di *B*. In caso affermativo diremo che il nastro è *bilanciato*, altrimenti diremo che il nastro è *sbilanciato*.

...	A	A	A	A	A	A	A	A	A	B	B	B	B	B	B	B	...
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----

Figura 1: Un esempio di nastro da analizzare.

Per rispondere a questa domanda però non potete contare le lettere: immaginate che il nastro sia molto lungo e che quindi diventi facile “saltare” una o più lettere. Per essere sicuri di non fare errori, utilizziamo il semplice marchingegno descritto in Figura 2, costituito da un cartoncino al cui interno è possibile inserire uno dei nastri da analizzare, in modo che una delle sue caselle sia evidenziata nella parte centrale.

Utilizzando questo marchingegno dovete trovare un procedimento che permetta di stabilire se un nastro sia o meno bilanciato. Il procedimento dovrà essere descritto come un insieme di regole, ognuna delle quali, in funzione della lettera contenuta nella casella evidenziata, consiste nell’effettuare due scelte:

- scegliere se cancellare o meno la lettera nella casella evidenziata (scrivendo sopra di essa una croce a penna);
- scegliere se lasciare il nastro nella sua posizione attuale, piuttosto che spostarlo a sinistra oppure a destra di una casella.

Dovrete quindi stabilire tale insieme di regole, in modo che posizionando il nastro così che risulti evidenziata la sua prima casella, e applicando queste regole fintantoché ciò risulti possibile, alla fine sia possibile stabilire quando un nastro è bilanciato e quando è sbilanciato.

Scrivete le regole su un foglio.

La macchina di Alan Turing

Alan M. Turing è uno dei padri fondatori dell’informatica moderna: nacque il 23 giugno 1912. In suo onore si intitola il più prestigioso premio alle scoperte informatiche, il Turing Award. Uno dei suoi contributi fondamentali è la proposta di una macchina ideale, la *Macchina di Turing*, inventata

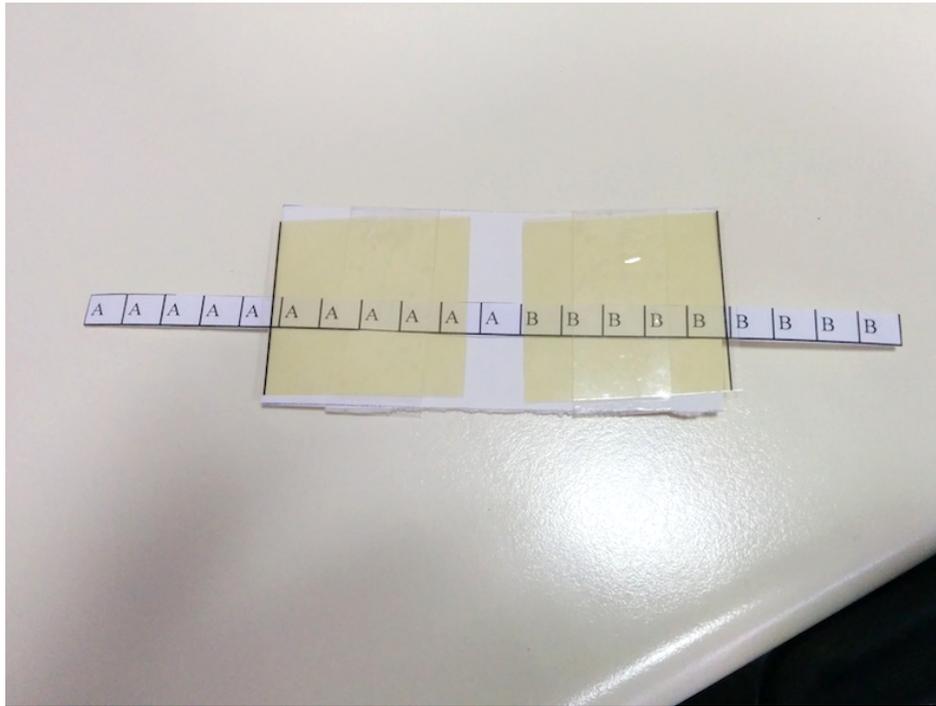


Figura 2: Il marchingegno che utilizzeremo.

per chiarire la nozione di computabilità, per cercare cioè di rispondere alla domanda: “cosa significa descrivere una elaborazione simbolica senza ambiguità”?

Il funzionamento della Macchina di Turing si basa su pochi concetti.

- Innanzitutto la macchina è dotata di un *nastro* composto da infinite caselle — ciascuna capace di ospitare una lettera; il nastro può scorrere verso destra o verso sinistra di una casella alla volta.
- Vi è poi un dispositivo, chiamato *testina*, in grado di leggere la lettera presente nella casella sulla quale è posizionato, e di sostituirla con una nuova lettera.
- In ogni momento la macchina si trova in un *stato* (inizialmente 0).
- Il funzionamento della macchina è dettato da un *programma*, ovvero un catalogo di *mosse*; ciascuna mossa determina, a partire dallo stato in cui si trova la macchina e dal simbolo letto dalla testina, qual è il prossimo stato in cui si troverà la macchina, come sostituire il simbolo letto, e in che direzione far eventualmente scorrere il nastro.

Più precisamente, una mossa è della forma

$$(S1, x) > (S2, y, a) \tag{1}$$

dove $S1$ e $S2$ sono stati, x e y sono lettere e a può valere s , d oppure $-$; la componente a descrive il movimento del nastro: s significa *a sinistra*, d significa *a destra*, $-$ significa *nessun movimento*. Il

significato di questa mossa è il seguente: “Se la macchina è nello stato $S1$ e la testina legge la lettera x , si passi nello stato $S2$, si scriva la lettera y al posto di x e si muova il nastro di una casella nella direzione indicata da a ”.

Una volta avviata, la macchina continua l'esecuzione del programma fintanto che esiste una mossa che riporta come primo elemento lo stato attuale della macchina e come secondo la lettera letta dalla testina. Se non c'è una mossa di questo tipo, la macchina si ferma.

Valgono le seguenti regole per l'inserimento di un programma:

- Ogni mossa va scritta su di una sola riga
- I nomi degli stati ($S1$ e $S2$ nell'esempio) possono essere composti solo dai caratteri ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.
- Le lettere che vengono lette e scritte sul nastro (x e y nell'esempio) possono essere solo ABCDEFGHIJKLMNOPQRSTUVWXYZ; la casella vuota si indica con il simbolo *
- Le azioni (a nell'esempio) sono uno dei caratteri s, d, -

Per esempio: l'istruzione $(0, *) > (1, C, s)$ significa: “Se la macchina è nello stato 0 e la testina legge una casella vuota, si passi nello stato 1, si scriva la lettera C e si muova il nastro di una casella verso sinistra”.

Un simulatore di Macchine di Turing

Collegandovi con un browser all'indirizzo

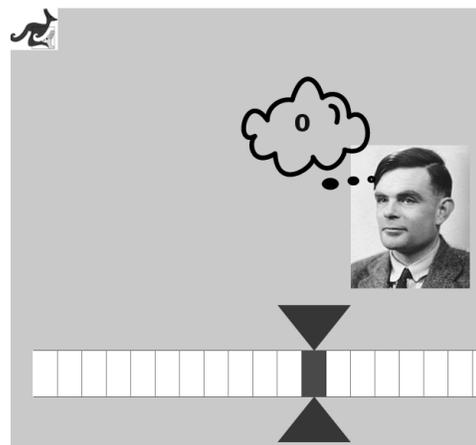
<http://kangourou.di.unimi.it/2012/turing/myturing.html>

potete utilizzare un semplice simulatore di Macchine di Turing.

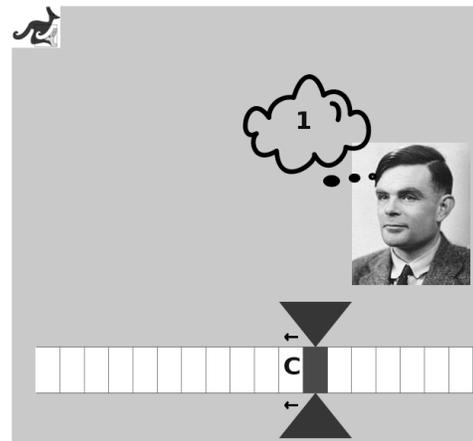
Seguiamo passo passo l'esecuzione del programma che segue con la Macchina di Turing, che è quello automaticamente caricato nel simulatore:

$(0, *) > (1, C, s)$
 $(1, *) > (2, I, s)$
 $(2, *) > (3, A, s)$
 $(3, *) > (4, 0, s)$

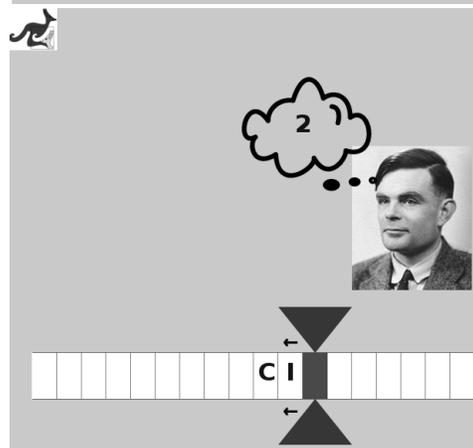
La situazione iniziale, con nastro vuoto, è la seguente:



A questo punto — essendo la macchina nello stato 0 e la casella letta dalla testina vuota — l'unica mossa applicabile è $(0, *) \rightarrow (1, C, s)$, che cambia lo stato della macchina in 1, scrive sul nastro C e sposta il nastro a sinistra di una casella.



A questo punto — essendo la macchina nello stato 1 e la casella letta dalla testina vuota — l'unica mossa applicabile è $(1, *) \rightarrow (2, I, s)$, che cambia lo stato della macchina in 2, scrive sul nastro I e sposta il nastro a sinistra di una casella.



A questo punto — essendo la macchina nello stato 2 e la casella letta dalla testina vuota — l'unica mossa applicabile è $(2, *) \rightarrow (3, A, s)$, che cambia lo stato della macchina in 3, scrive sul nastro A e sposta il nastro a sinistra di una casella.

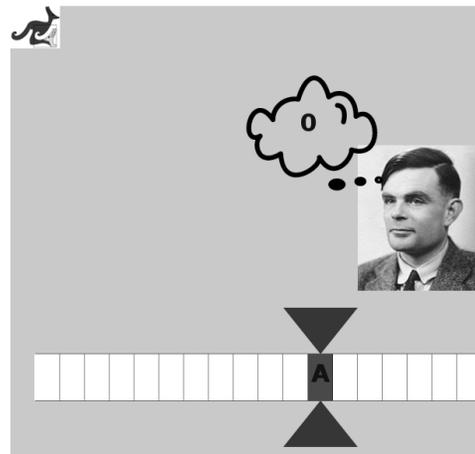


A questo punto — essendo la macchina nello stato 3 e la casella letta dalla testina vuota — l'unica mossa applicabile è $(3, *) \rightarrow (4, 0, s)$, che cambia lo stato della macchina in 4, scrive sul nastro 0 e sposta il nastro a sinistra di una casella.



A questo punto non c'è più alcuna mossa che risulti applicabile e l'esecuzione del programma termina.

Si noti che qualora inizialmente la testina fosse posizionata su di una casella contenente una lettera, il programma non avrebbe potuto essere eseguito, perché non c'è alcuna mossa attivabile nello stato 0 e testina che legge una casella diversa dalla casella vuota.



Esercizi

1. Cosa fa il seguente programma eseguibile dalla Macchina di Turing quando sul nastro di partenza c'è una sequenza di A, B e C (per esempio AAAABC o ABBBBBB)?

$(0, A) \rightarrow (0, *, s)$
 $(0, B) \rightarrow (0, *, s)$
 $(0, C) \rightarrow (1, *, s)$
 $(1, *) \rightarrow (2, V, -)$
 $(1, A) \rightarrow (1, *, s)$
 $(1, B) \rightarrow (1, *, s)$
 $(1, C) \rightarrow (1, *, s)$



2. Cosa fa il seguente programma eseguibile dalla Macchina di Turing quando sul nastro di partenza c'è una sequenza di A e B (per esempio AAAABB o AABB o A)? Il programma è caricabile nel simulatore scegliendo la voce di menu **Mistero**.

(0,A) > (1,*,s)
(1,A) > (1,A,s)
(1,B) > (1,B,s)
(1,*) > (2,*,d)
(2,B) > (3,*,d)
(3,A) > (3,A,d)
(3,B) > (3,B,d)
(3,*) > (0,*,s)
(0,*) > (4,S,s)
(4,*) > (F,I,s)
(0,B) > (F,*,-)



3. Programmate la Macchina di Turing in modo che, data sul nastro di partenza una sequenza qualsiasi delle lettere 0 e I, aggiunga una lettera I se il numero delle I già presente è dispari, altrimenti aggiunga una 0: in ogni caso, quindi, il numero finale di I deve essere pari.

nastro iniziale	nastro finale
I0	IOI
IOI	IOIO
0	00



4. Programmate la Macchina di Turing in modo che, dato un nastro iniziale contenente una sequenza qualsiasi di A,B, e C, termina la sua esecuzione lasciando sul nastro la sequenza che si ottiene eliminando tutte le A iniziali fino alla prima lettera diversa da A più a sinistra. Qualora la sequenza iniziale sia composta da sole A, il risultato finale è A.

nastro iniziale	nastro finale
ABC	BC
ABAC	BAC
BBB	BBB
AAAA	A

