

Il calcolatore universale, origini e nuovi paradigmi

Lucia Pomello

Università degli studi di Milano–Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione

24 febbraio 2017



L'evoluzione dell'informatica

Prima (fino agli anni '40)



Calcolatori umani

Computing Division
Dipartimento del Tesoro,
Washington, 1924

Ora

Computazione distribuita



Due grandi sogni

- una macchina automatica per fare i calcoli

Due grandi sogni

- una macchina automatica per fare i calcoli
- *il sogno di [Leibniz](#) (1646 - 1716): "definire un insieme di simboli e regole per formalizzare il ragionamento logico", nasce la Logica Simbolica ("L'Arte Combinatoria" – 1666)*

I fondamenti: la teoria della computabilità

- Parigi 1900 - Il Congresso Internazionale di Matematica
David Hilbert: *i 23 problemi aperti del millennio*

I fondamenti: la teoria della computabilità

- Parigi 1900 - Il Congresso Internazionale di Matematica
David Hilbert: *i 23 problemi aperti del millennio*
- 1928 David Hilbert: “Esiste una procedura effettiva per decidere se una formula è vera?”
(*Entscheidungsproblem*, il problema della decisione)

I fondamenti: la teoria della computabilità

- Parigi 1900 - Il Congresso Internazionale di Matematica
David Hilbert: *i 23 problemi aperti del millennio*
- 1928 **David Hilbert:** “*Esiste una procedura effettiva per decidere se una formula è vera?*”
(*Entscheidungsproblem*, il problema della decisione)
- 1936 tre risposte indipendenti ed equivalenti
 - **Alan M. Turing:** la macchina di Turing
 - **Alonzo Church:** λ -calcolo
 - **Kurt Gödel:** le funzioni (primitive) ricorsive

funzione $f : \mathbb{N} \rightarrow \mathbb{N}$

esempio: *doppio*

n	2n
0	0
1	2
2	4
3	6
4	8
5	10
6	12
7	14
8	16

funzione $f : \mathbb{N} \rightarrow \mathbb{N}$

doppio è computabile

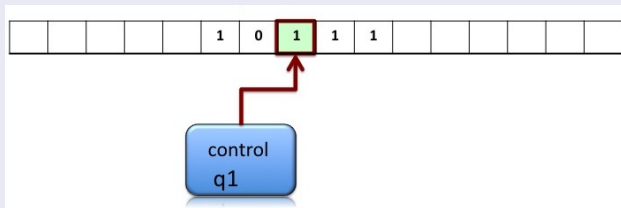
n	2n
0	0
1	2
2	4
-	-
-	-
-	-
-	-
107	

On Computable Numbers, with an Application to the Entscheidungsproblem, 1936

La macchina di Turing



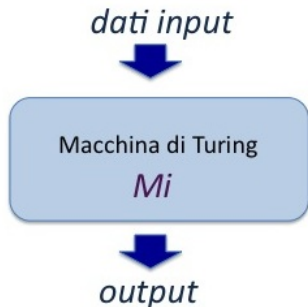
Alan M. Turing
(1912-1954)



Componenti:

- nastro, testina di lettura/scrittura
- insieme finito di simboli (es: $\{0, 1\}$)
- controllo:
 - * insieme finito di stati (es: $\{q_0, q_1, q_2, q_3\}$)
 - * regole di cambiamento di stato

funzioni
 $f : \mathbb{N} \rightarrow \mathbb{N}$

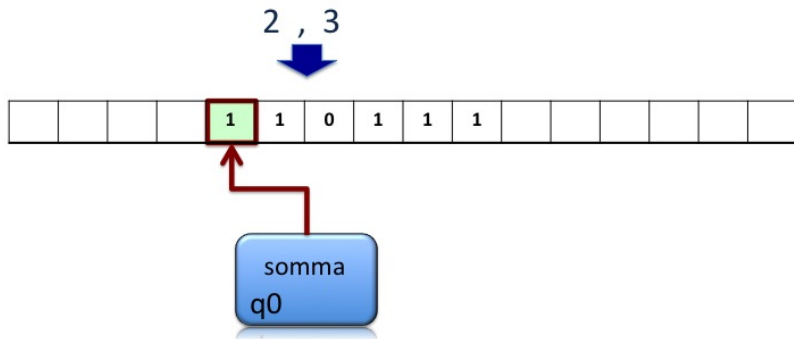


Tesi di Turing - Church

una funzione f è *computabile*
se e solo se
esiste una Macchina di Turing che *calcola* f

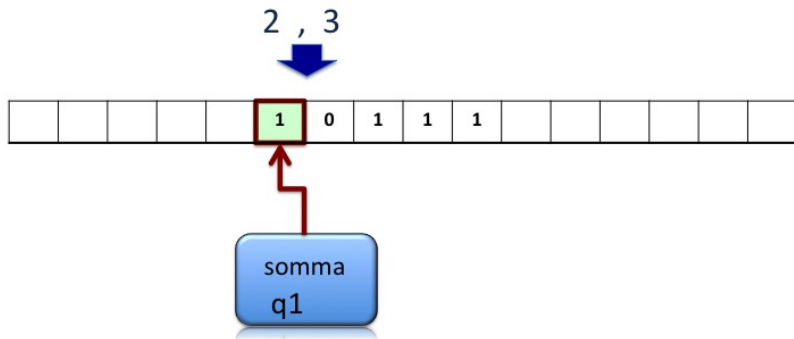
La macchina di Turing: un esempio, la funzione somma

Somma(2, 3)



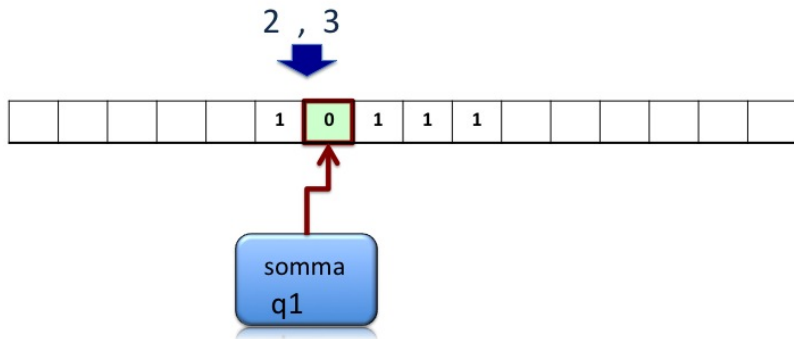
La macchina di Turing: un esempio

Somma(2, 3)



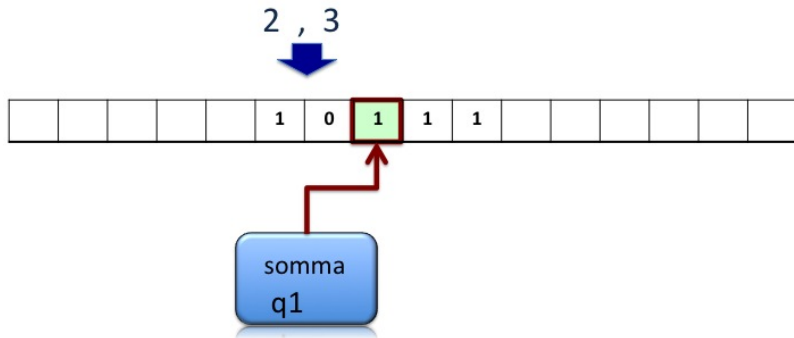
La macchina di Turing: un esempio

Somma(2, 3)



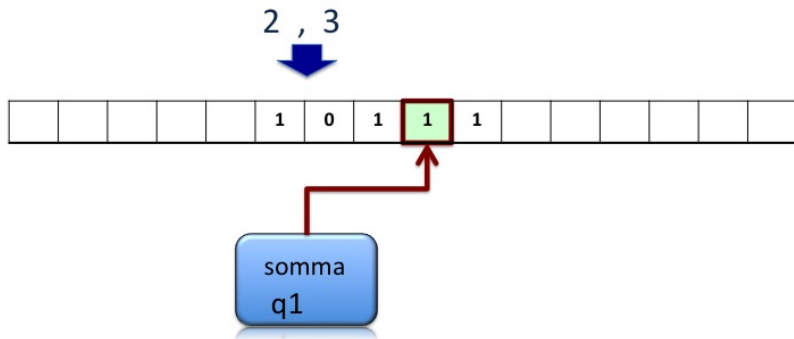
La macchina di Turing: un esempio

Somma(2, 3)



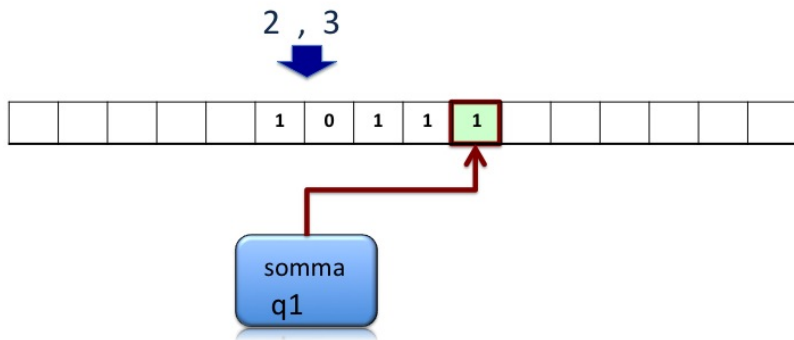
La macchina di Turing: un esempio

Somma(2, 3)



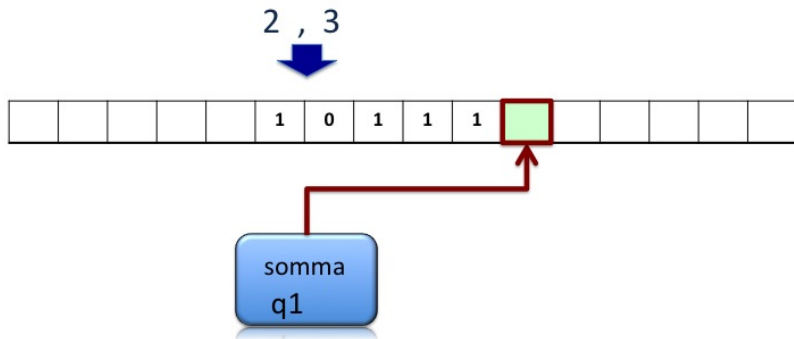
La macchina di Turing: un esempio

Somma(2, 3)



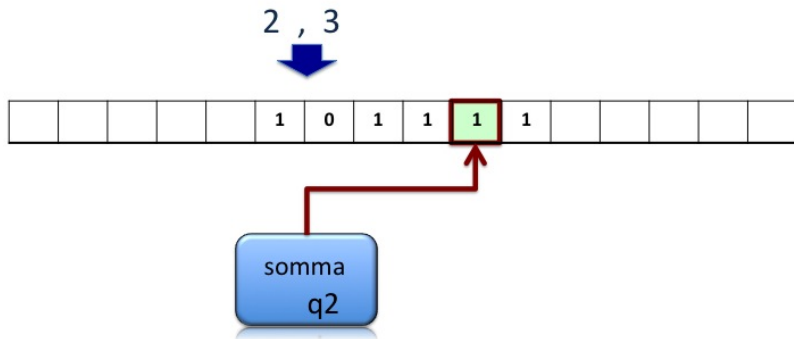
La macchina di Turing: un esempio

Somma(2, 3)



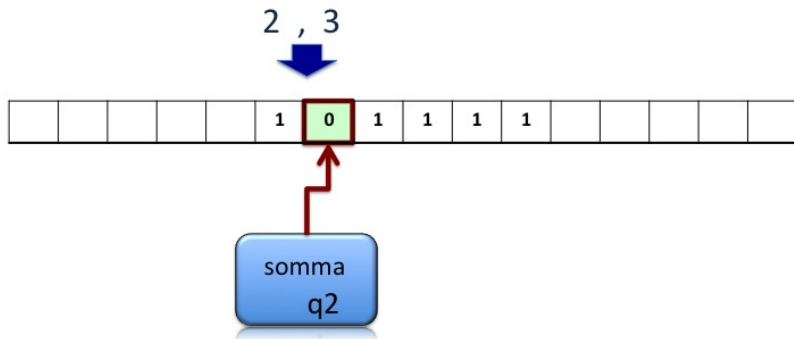
La macchina di Turing: un esempio

Somma(2, 3)



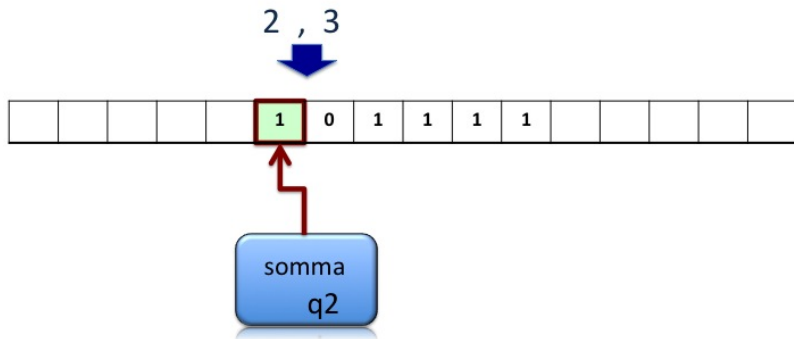
La macchina di Turing: un esempio

Somma(2, 3)



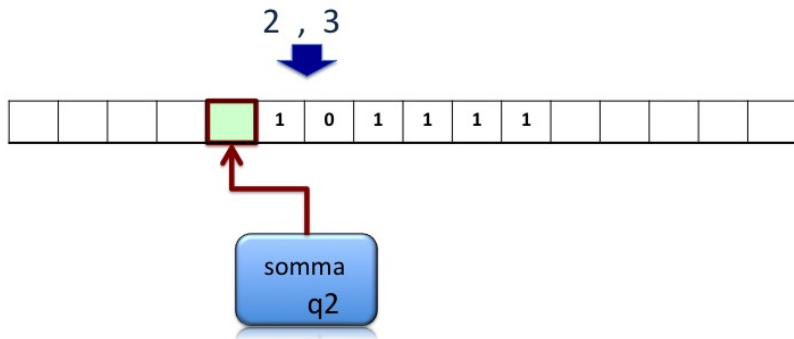
La macchina di Turing: un esempio

Somma(2, 3)



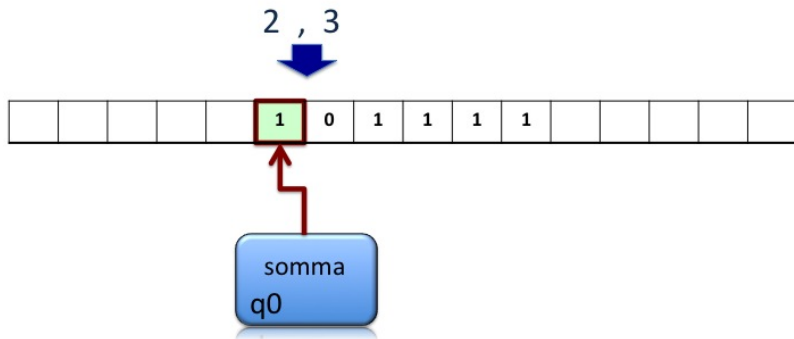
La macchina di Turing: un esempio

Somma(2, 3)



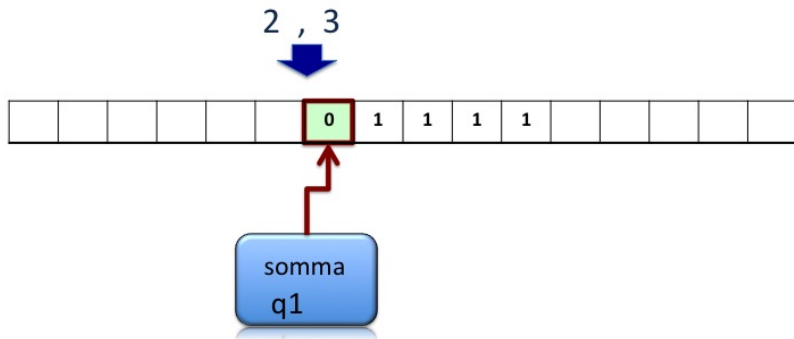
La macchina di Turing: un esempio

Somma(2, 3)



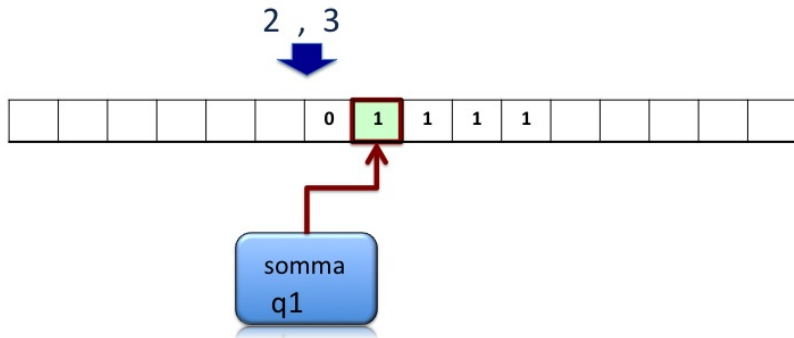
La macchina di Turing: un esempio

Somma(2, 3)



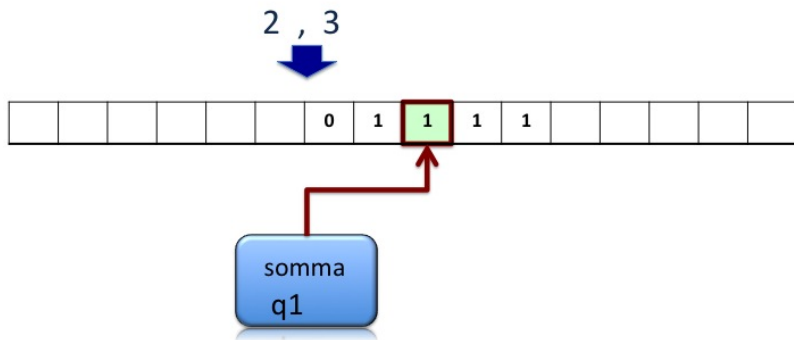
La macchina di Turing: un esempio

Somma(2, 3)



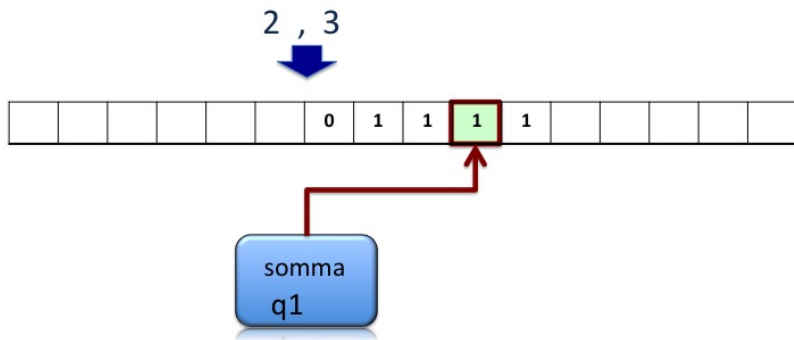
La macchina di Turing: un esempio

Somma(2, 3)



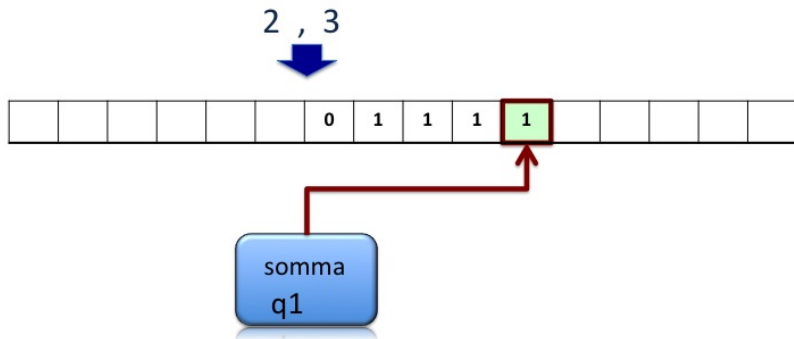
La macchina di Turing: un esempio

Somma(2, 3)



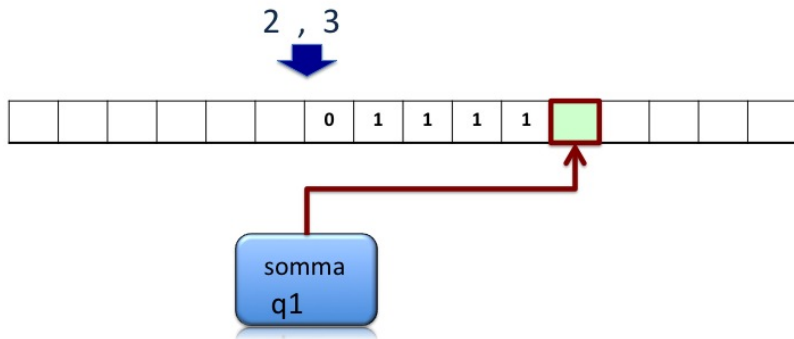
La macchina di Turing: un esempio

Somma(2, 3)



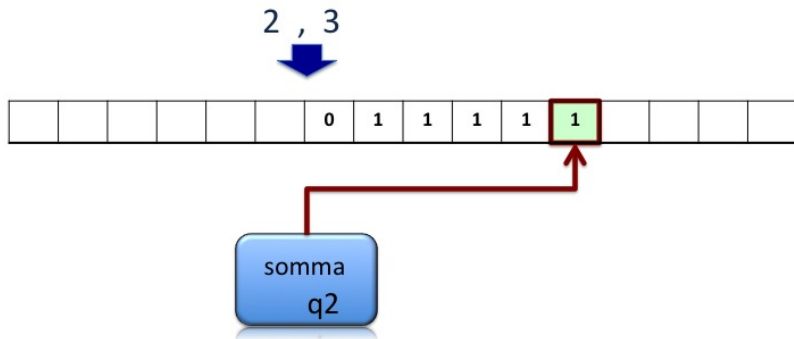
La macchina di Turing: un esempio

Somma(2, 3)



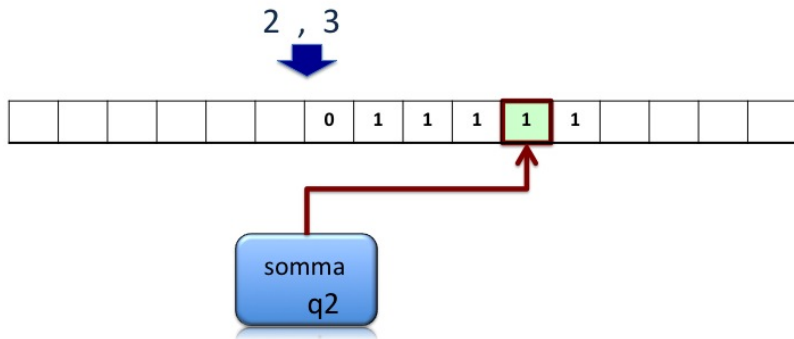
La macchina di Turing: un esempio

Somma(2, 3)



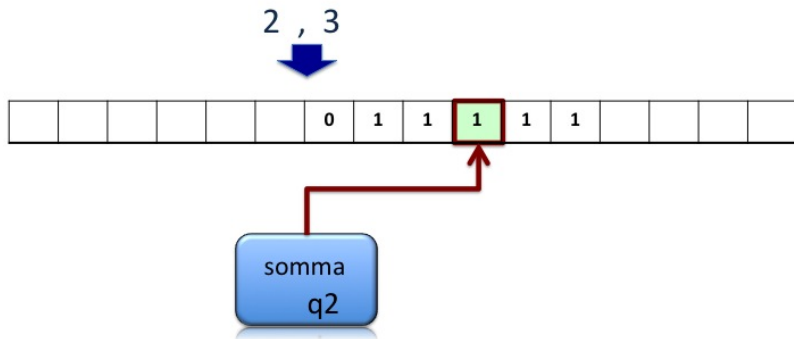
La macchina di Turing: un esempio

Somma(2, 3)



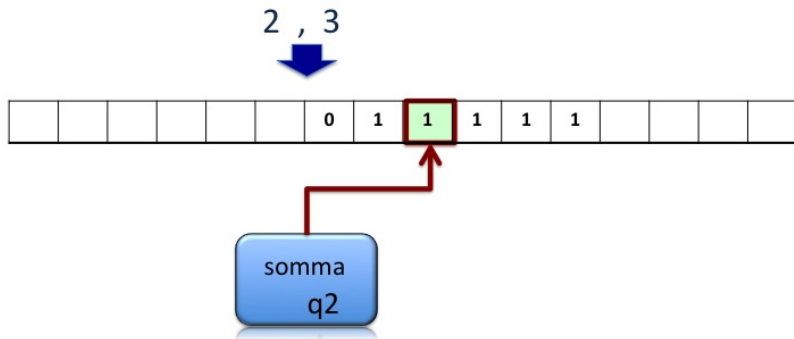
La macchina di Turing: un esempio

Somma(2, 3)



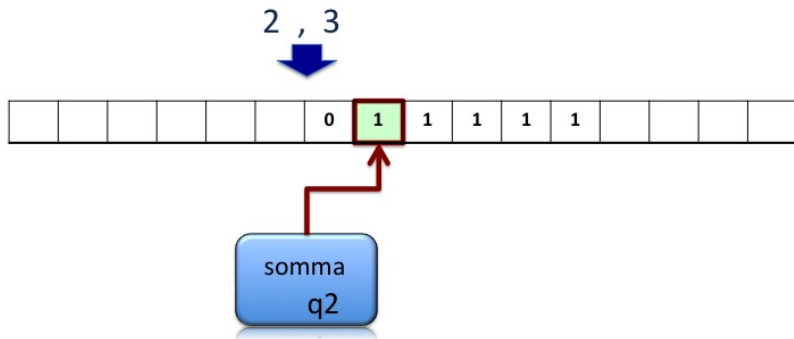
La macchina di Turing: un esempio

Somma(2, 3)



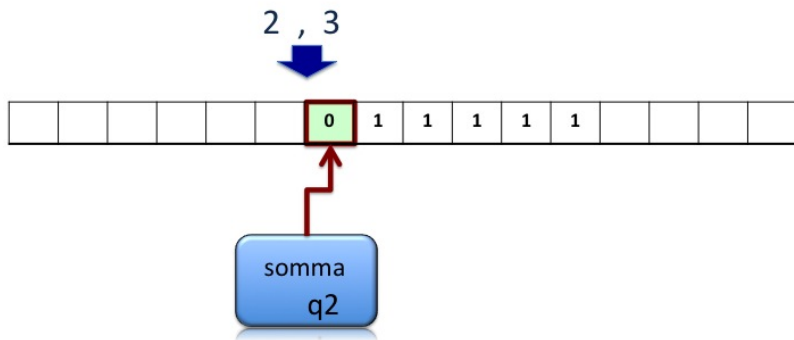
La macchina di Turing: un esempio

Somma(2, 3)



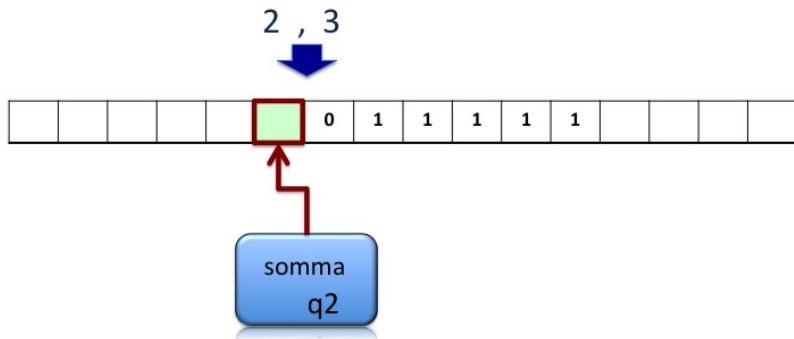
La macchina di Turing: un esempio

Somma(2, 3)



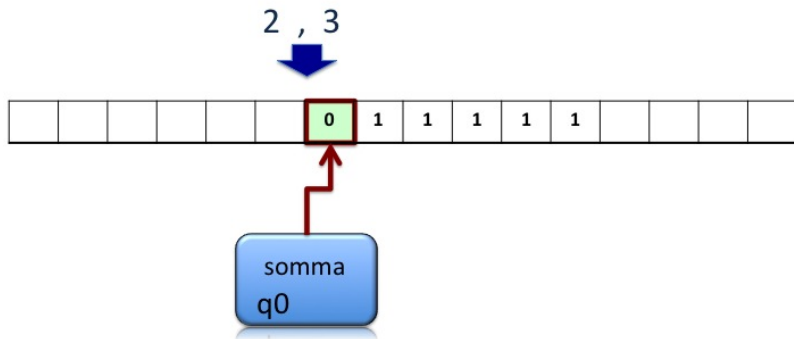
La macchina di Turing: un esempio

Somma(2, 3)



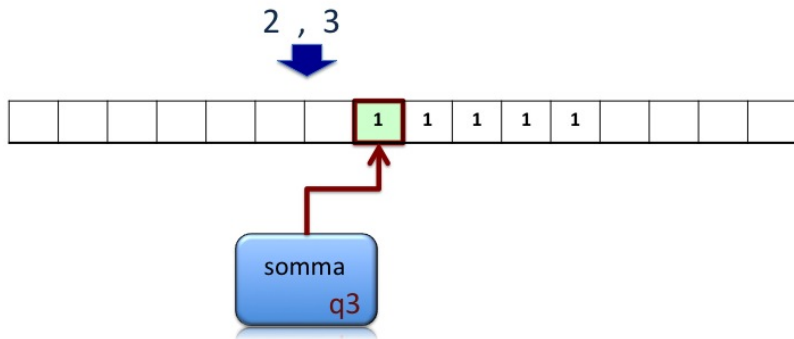
La macchina di Turing: un esempio

Somma(2, 3)



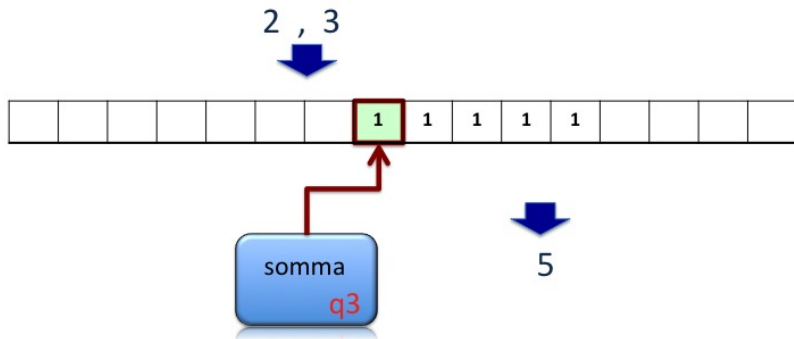
La macchina di Turing: un esempio

Somma(2, 3)

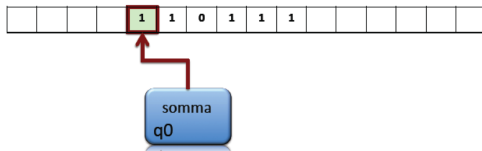


La macchina di Turing: un esempio

Somma(2, 3)

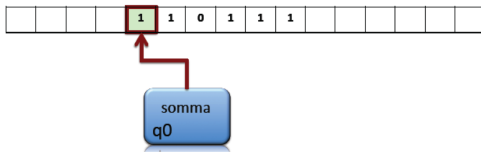


La macchina di Turing: rappresentazione e codifica



Somma		0	1
q0		q3 →	q1 →
q1	q2 1 ←	q1 0 →	q1 1 →
q2	q0 →	q2 0 ←	q2 1 ←
q3			

La macchina di Turing: rappresentazione e codifica

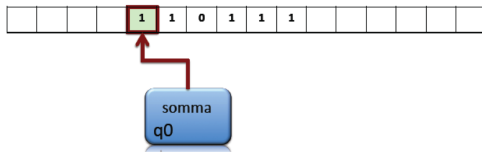


Somma		0	1
q0		q3 →	q1 →
q1	q2 1 ←	q1 0 →	q1 1 →
q2	q0 →	q2 0 ←	q2 1 ←
q3			

codifica

ogni macchina di Turing M_i può essere *codificata* da una sequenza di caratteri (binari), da un numero naturale i .

La macchina di Turing: rappresentazione e codifica



Somma		0	1
q0		q3 →	q1 →
q1	q2 1 ←	q1 0 →	q1 1 →
q2	q0 →	q2 0 ←	q2 1 ←
q3			

codifica

ogni macchina di Turing M_i può essere *codificata* da una sequenza di caratteri (binari), da un numero naturale i .

⇒ due conseguenze fondamentali

1. La macchina di Turing universale

"We do not need to have an infinity of different machines doing different jobs. A single one will suffice."

[A. M. Turing, 1948]

1. La macchina di Turing universale

$\langle \boxed{M_i}, \text{dati input} \rangle$



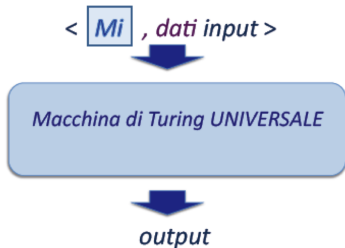
Macchina di Turing UNIVERSALE



output

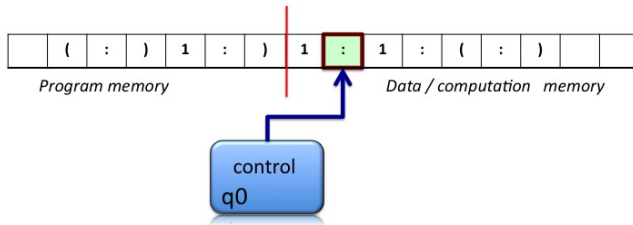
simula la computazione di una qualsiasi altra macchina M_i

1. La macchina di Turing universale

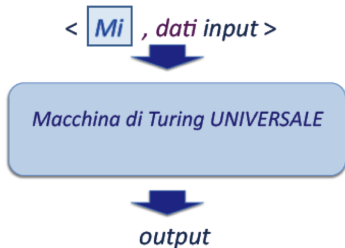


simula la computazione di una qualsiasi altra macchina M_i

La macchina universale

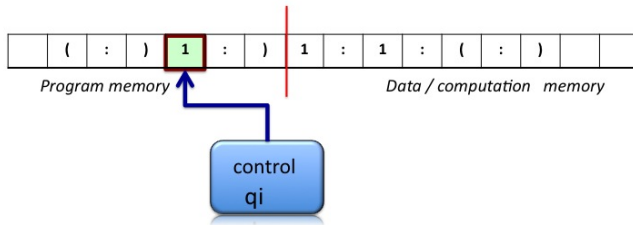


1. La macchina di Turing universale



simula la computazione di una qualsiasi altra macchina

La macchina universale



2. Funzioni non computabili

- ogni macchina di Turing M_i *codificata* da un numero naturale i
- ogni macchina di Turing M_i calcola una funzione f_i
(Tesi di Turing-Church)

2. Funzioni non computabili

- ogni macchina di Turing M_i *codificata* da un numero naturale i
- ogni macchina di Turing M_i calcola una funzione f_i
(Tesi di Turing-Church)
- ci sono molte più funzioni dei numeri naturali



2. Funzioni non computabili

- ogni macchina di Turing M_i *codificata* da un numero naturale i
- ogni macchina di Turing M_i calcola una funzione f_i
(Tesi di Turing-Church)
- ci sono molte più funzioni dei numeri naturali



Teorema [Turing, 1936]

Esistono funzioni *non computabili* (problemi non decidibili)

2. Funzioni non computabili

- ogni macchina di Turing M_i *codificata* da un numero naturale i
- ogni macchina di Turing M_i calcola una funzione f_i
(Tesi di Turing-Church)
- ci sono molte più funzioni dei numeri naturali



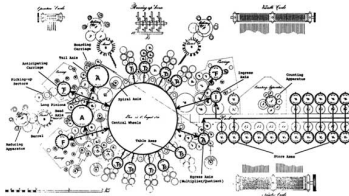
Teorema [Turing, 1936]

Esistono funzioni *non computabili* (problemi non decidibili)

Un esempio di problema non decidibile

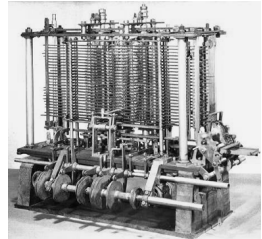
- il problema dell'ALT

La prima macchina programmabile

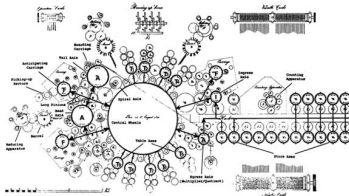


il progetto della
Macchina analitica – 1834

Charles Babbage (1791 – 1871)

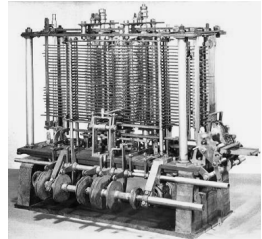


La prima macchina programmabile



il progetto della
Macchina analitica – 1834

Charles Babbage (1791 – 1871)



Ada Byron (1815 – 1851)
il primo programma

Dalle macchine astratte ai calcolatori concreti

Konrad Zuse: Z1 - Z4

Z1 - Berlino 1936 - '38



1^a macchina con *codice binario* e *programmabile*

Z4 - Zurigo 1950



1^a macchina digitale *elettro-meccanica commerciale*

Plankalkül (1943 – 1945) il primo linguaggio di programmazione

Dalle macchine astratte ai calcolatori concreti

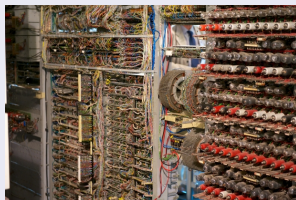
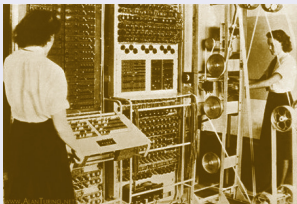
Bletchley Park - II Guerra mondiale



Bomba simula Enigma



Colossus 1943 – '44



Progettata da Tommy Flowers, "la prima macchina elettronica" programmabile, usata per decifrare i messaggi tra Hitler e i suoi generali (criptati con la macchina Tunny).

Oltre Turing: nuovi paradigmi

Prima

- sistema chiuso
(Input $\longrightarrow$ Output)
- computazione
sequenziale
- terminante

Ora

- sistema aperto
(componenti interagenti)
- computazione
parallela/distribuita
- non necessariamente terminante

Computazione interattiva

- Computazione parallela / distribuita
- Reti di computazione e di comunicazione
(es: Mobile computing, Cloud computing, GRID computing)
- Sistemi di supporto al lavoro cooperativo, . . .

Computazione interattiva

- Computazione parallela / distribuita
- Reti di computazione e di comunicazione
(es: Mobile computing, Cloud computing, GRID computing)
- Sistemi di supporto al lavoro cooperativo, . . .

Concetti fondamentali

- Competizione, risorse condivise
- Cooperazione, distribuzione
- Sincronizzazione
- Flusso di informazione
- Sicurezza, privacy

Computazione interattiva

Modelli

- Reti di Petri
- Automi cellulari
- . . .

Computazione interattiva

Modelli

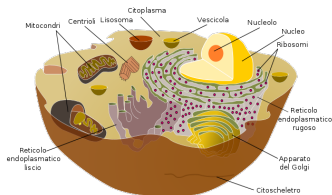
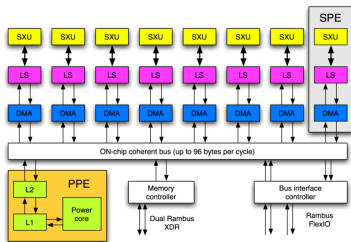
- Reti di Petri
- Automi cellulari
- . . .

Come gestire la complessità di questi sistemi?

Tecniche formali:

- di costruzione del modello e di implementazione
- di verifica delle proprietà di comportamento
(astrazione - modularità - composizionalità)

Oltre Turing: nuovi paradigmi



Natural computation

- *Simulazioni di fenomeni naturali*
 - Biologici
 - Fisici
 - Geo-fisici, ...

Natural computation

- *Simulazioni di fenomeni naturali*
 - Biologici
 - Fisici
 - Geo-fisici, ...
- *Computazioni che si ispirano alla natura*
 - Reti neurali
 - Computazione a membrane
 - ...

Natural computation

- *Simulazioni di fenomeni naturali*
 - Biologici
 - Fisici
 - Geo-fisici, ...
- *Computazioni che si ispirano alla natura*
 - Reti neurali
 - Computazione a membrane
 - ...
- *Computazione in natura*
 - Molecular computing
 - DNA computing
 - Computazione quantistica
 - ...

Grazie dell'attenzione!!

