Che cos'è la computazione

Gianluca Curzi

Università degli Studi di Urbino
Laboratorio PLS (Piano Lauree Scientifiche)

28 novembre 2019

Outline

Cosa significa calcolare

Cosa significa calcolare? Cos'è un algoritmo?

Partiamo da un esempio conosciuto:

$$
\begin{array}{r}
4231\times \\
77 \\
\hline
29617 \\
29617 \\
\hline
325787
\end{array}
$$

Cosa significa calcolare? Cos'è un algoritmo?

Siamo interessati a sapere cosa significa calcolare "meccanicamente", ossia *algoritmicamente*:

Algoritmo

Un algoritmo è un insieme finito di istruzioni chiare e non ambigue che permette di risolvere un problema.

Esempi

I testi di studio di matematica delle superiori sono pieni di algoritmi:

- **Elementari**: addizione, moltiplicazione, sottrazione, divisione.
- **Medie**: risolvere equazioni di un certo grado manipolando espressioni algebriche.
- **Superiori**: calcolo infinitesimale (risolvere limiti).

Cosa significa calcolare? Cos'è un algoritmo?

Siamo interessati a sapere cosa significa calcolare "meccanicamente", ossia *algoritmicamente*:

Algoritmo

Un algoritmo è un insieme finito di istruzioni chiare e non ambigue che permette di risolvere un problema.

Esempi

I testi di studio di matematica delle superiori sono pieni di algoritmi:

- **Elementari**: addizione, moltiplicazione, sottrazione, divisione.
- **Medie**: risolvere equazioni di un certo grado manipolando espressioni algebriche.
- **Superiori**: calcolo infinitesimale (risolvere limiti!).

Cosa significa calcolare? Cos'è un algoritmo?

Siamo interessati a sapere cosa significa calcolare "meccanicamente", ossia *algoritmicamente*:

Algoritmo

Un algoritmo è un insieme finito di istruzioni chiare e non ambigue che permette di risolvere un problema.

Esempi

I testi di studio di matematica delle superiori sono pieni di algoritmi:

- **Elementari**: addizione, moltiplicazione, sottrazione, divisione.
- **Medie**: risolvere equazioni di un certo grado manipolando espressioni algebriche.
- **Superiori**: calcolo infinitesimale (risolvere limiti!).

Cosa significa calcolare? Cos'è un algoritmo?

Siamo interessati a sapere cosa significa calcolare "meccanicamente", ossia *algoritmicamente*:

Algoritmo

Un algoritmo è un insieme finito di istruzioni chiare e non ambigue che permette di risolvere un problema.

Esempi

I testi di studio di matematica delle superiori sono pieni di algoritmi:

- **Elementari**: addizione, moltiplicazione, sottrazione, divisione.
- **Medie**: risolvere equazioni di un certo grado manipolando espressioni algebriche.
- **Superiori**: calcolo infinitesimale (risolvere limiti!).

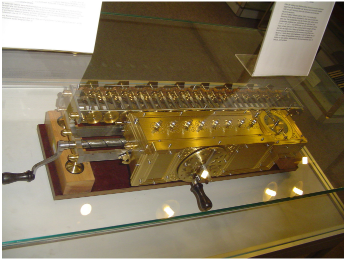
Un po' di storia

Leibniz

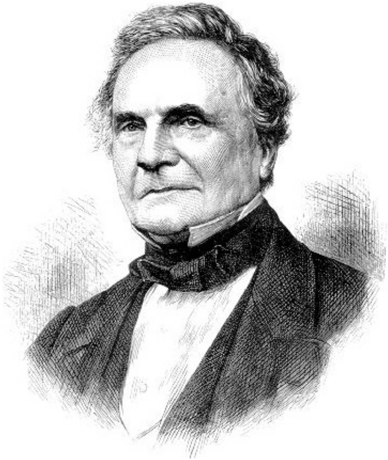
- Gottfried Leibniz nacque a Lipsia nel 1646. E' stato un influente matematico e filosofo.
- Nel 1673 presenta un modello di macchina capace di eseguire le quattro operazioni aritmetiche fondamentali (addizione, moltiplicazione, sottrazione, divisione). Uno dei precursori delle moderne calcolatrici.
- La sua macchina poteva fare solo aritmetica ordinaria. Ma Leibniz intuiva che riuscire a meccanizzare il calcolo aveva un significato più ampio.

- Gottfried Leibniz nacque a Lipsia nel 1646. E' stato un influente matematico e filosofo.
- Nel 1673 presenta un modello di macchina capace di eseguire le quattro operazioni aritmetiche fondamentali (addizione, moltiplicazione, sottrazione, divisione). Uno dei precursori delle moderne calcolatrici.
- La sua macchina poteva fare solo aritmetica ordinaria. Ma Leibniz intuiva che riuscire a meccanizzare il calcolo aveva un significato più ampio.

- Gottfried Leibniz nacque a Lipsia nel 1646. E' stato un influente matematico e filosofo.
- Nel 1673 presenta un modello di macchina capace di eseguire le quattro operazioni aritmetiche fondamentali (addizione, moltiplicazione, sottrazione, divisione). Uno dei precursori delle moderne calcolatrici.
- La sua macchina poteva fare solo aritmetica ordinaria. Ma Leibniz intuiva che riuscire a meccanizzare il calcolo aveva un significato più ampio.



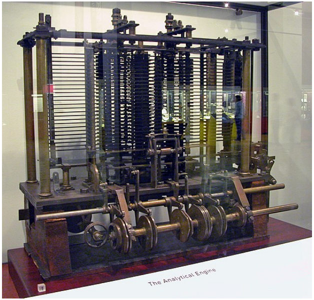
Babbage



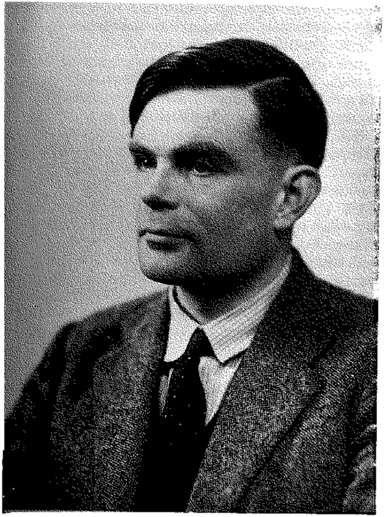
- Charles Babbage nacque a Londra nel 1791. E' stato un matematico e filosofo.
- Nel 1833 progetta la *macchina analitica*, una macchina capace di effettuare complessi calcoli matematici.
- La macchina analitica rappresenta il primo vero calcolatore programmabile, ma non fu mai realizzata.

- Charles Babbage nacque a Londra nel 1791. E' stato un matematico e filosofo.
- Nel 1833 progetta la *macchina analitica*, una macchina capace di effettuare complessi calcoli matematici.
- La macchina analitica rappresenta il primo vero calcolatore programmabile, ma non fu mai realizzata.

- Charles Babbage nacque a Londra nel 1791. E' stato un matematico e filosofo.
- Nel 1833 progetta la *macchina analitica*, una macchina capace di effettuare complessi calcoli matematici.
- La macchina analitica rappresenta il primo vero calcolatore programmabile, ma non fu mai realizzata.



Turing



- Alan Mathison Turing nacque a Londra il 23 giugno 1912.
- talento matematico non comune: già da ragazzo faceva ricerche matematiche per proprio conto e. . . studiava la relatività di Einstein.
- Nel 1931 fu ammesso al King's College di Cambridge e si laureò col massimo dei voti.
- Nel 1936 Turing pubblicò "*On Computable Numbers, with an Application to the Entscheidungsproblem*". E' l'articolo in cui compare per la prima volta la famosa "macchina di Turing".

- Alan Mathison Turing nacque a Londra il 23 giugno 1912.
- talento matematico non comune: già da ragazzo faceva ricerche matematiche per proprio conto e. . . studiava la relatività di Einstein.
- Nel 1931 fu ammesso al King's College di Cambridge e si laureò col massimo dei voti.
- Nel 1936 Turing pubblicò "*On Computable Numbers, with an Application to the Entscheidungsproblem*". E' l'articolo in cui compare per la prima volta la famosa "macchina di Turing".

- Alan Mathison Turing nacque a Londra il 23 giugno 1912.
- talento matematico non comune: già da ragazzo faceva ricerche matematiche per proprio conto e. . . studiava la relatività di Einstein.
- Nel 1931 fu ammesso al King's College di Cambridge e si laureò col massimo dei voti.
- Nel 1936 Turing pubblicò "*On Computable Numbers, with an Application to the Entscheidungsproblem*". E' l'articolo in cui compare per la prima volta la famosa "macchina di Turing".

- Alan Mathison Turing nacque a Londra il 23 giugno 1912.
- talento matematico non comune: già da ragazzo faceva ricerche matematiche per proprio conto e. . . studiava la relatività di Einstein.
- Nel 1931 fu ammesso al King's College di Cambridge e si laureò col massimo dei voti.
- Nel 1936 Turing pubblicò "*On Computable Numbers, with an Application to the Entscheidungsproblem*". E' l'articolo in cui compare per la prima volta la famosa "macchina di Turing".

La macchina di Turing

Turing analizza il processo di calcolo

- Turing non vuole costruire macchine concrete per fare specifici calcoli, come fanno Leibniz e Babbage. Turing vuole dare una definizione *astratta* e *formale* di che cos'è un processo di calcolo.
- Si concentra su cosa fa esattamente una *persona* quando calcola.
- Elimina uno dopo l'altro i *dettagli inessenziali*.

Turing analizza il processo di calcolo

- Turing non vuole costruire macchine concrete per fare specifici calcoli, come fanno Leibniz e Babbage. Turing vuole dare una definizione *astratta* e *formale* di che cos'è un processo di calcolo.
- Si concentra su cosa fa esattamente una *persona* quando calcola.
- Elimina uno dopo l'altro i *dettagli inessenziali*.

Turing analizza il processo di calcolo

- Turing non vuole costruire macchine concrete per fare specifici calcoli, come fanno Leibniz e Babbage. Turing vuole dare una definizione *astratta* e *formale* di che cos'è un processo di calcolo.
- Si concentra su cosa fa esattamente una *persona* quando calcola.
- Elimina uno dopo l'altro i *dettagli inessenziali*.

Supponiamo in concreto che una persona, diciamo Carlo, sia impegnato a calcolare 4231×77. Carlo traccerà segni su un foglio, spostando via via l'attenzione da quello che ha già scritto a quello che sta scrivendo.
Dobbiamo togliere il *superfluo*: Carlo sta prendendo un caffè mentre lavora? Scrive a matita o a penna? Le dimensioni dei fogli su cui Carlo scrive?
Possiamo immaginare, senza che niente di essenziale vada perduto, che Carlo lavori con un nastro di carta diviso in quadretti:

Supponiamo in concreto che una persona, diciamo Carlo, sia impegnato a calcolare 4231×77. Carlo traccerà segni su un foglio, spostando via via l'attenzione da quello che ha già scritto a quello che sta scrivendo.

Dobbiamo togliere il *superfluo*: Carlo sta prendendo un caffè mentre lavora? Scrive a matita o a penna? Le dimensioni dei fogli su cui Carlo scrive?

Possiamo immaginare, senza che niente di essenziale vada perduto, che Carlo lavori con un nastro di carta diviso in quadretti:

$$
\begin{array}{r}
4231\times \\
77 \\
\hline
29617 \\
29617 \\
\hline
325787
\end{array}
$$

Supponiamo in concreto che una persona, diciamo Carlo, sia impegnato a calcolare 4231×77. Carlo traccerà segni su un foglio, spostando via via l'attenzione da quello che ha già scritto a quello che sta scrivendo.
Dobbiamo togliere il *superfluo*: Carlo sta prendendo un caffè mentre lavora? Scrive a matita o a penna? Le dimensioni dei fogli su cui Carlo scrive?
Possiamo immaginare, senza che niente di essenziale vada perduto, che Carlo lavori con un nastro di carta diviso in quadretti:

$$\begin{array}{r} 4231\times \\ 77 \\ \hline 29617 \\ 29617 \\ \hline 325787 \end{array}$$

Supponiamo in concreto che una persona, diciamo Carlo, sia impegnato a calcolare 4231×77. Carlo traccerà segni su un foglio, spostando via via l'attenzione da quello che ha già scritto a quello che sta scrivendo.
Dobbiamo togliere il *superfluo*: Carlo sta prendendo un caffè mentre lavora? Scrive a matita o a penna? Le dimensioni dei fogli su cui Carlo scrive?
Possiamo immaginare, senza che niente di essenziale vada perduto, che Carlo lavori con un nastro di carta diviso in quadretti:

$$\begin{array}{r} 4231\times \\ 77 \\ \hline 29617 \\ 29617 \\ \hline 325787 \end{array}$$

4	2	3	1	×	7	7	=	2	9	6	1	7	+	2	9	6	1	7	0	=	3	2	5	7	8	7

4	2	3	1	×	7	7	=	2	9	6	1	7	+	2	9	6	1	7	0	=	3	2	5	7	8	7

Prima osservazione: il nastro

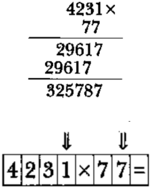
Turing si rende conto che ogni calcolo può essere eseguito in un nastro unidimensionale potenzialmente infinito e diviso in celle, ognuna delle quali può contenere al più un simbolo.

Inizialmente il nastro conterrà solo i due numeri che vogliamo moltiplicare. Indichiamo con una freccia $\Downarrow$ in quali caselle è focalizzata l'"attenzione" di Carlo, un po' come quando si usa un dito per tenere il segno. Man mano che Carlo calcola, la sua attenzione si sposterà di casella in casella: in alcune caselle scriverà dei simboli e a volte ne cancellerà qualcuno, così da lasciare spazio ad altri simboli.

Carlo sa che deve moltiplicare 1×7, per cui ottiene un 7 e lo scrive sul nastro. Ora sposta l'attenzione sulle cifre 3 e 7, che a loro volta vengono moltiplicate.

Dopo aver completato la fase di calcolo in cui moltiplica le cifre a due a due, dovrà sommare i due prodotti parziali. Carlo inizia sommando 7 e 0. Ora deve sommare 1 e 7 e otterrà 8.

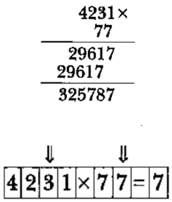
Notate che Carlo ha spostato due volte la sua attenzione sulle cifre 1 e 7, ma la prima volta le ha *moltiplicate* e la seconda le ha *sommate* perché il suo "stato mentale" è cambiato.



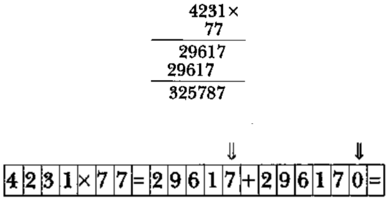
Inizialmente il nastro conterrà solo i due numeri che vogliamo moltiplicare. Indichiamo con una freccia ⇓ in quali caselle è focalizzata l'"attenzione" di Carlo, un po' come quando si usa un dito per tenere il segno. Man mano che Carlo calcola, la sua attenzione si sposterà di casella in casella: in alcune caselle scriverà dei simboli e a volte ne cancellerà qualcuno, così da lasciare spazio ad altri simboli. Carlo sa che deve moltiplicare 1×7, per cui ottiene un 7 e lo scrive sul nastro. Ora sposta l'attenzione sulle cifre 3 e 7, che a loro volta vengono moltiplicate.

Dopo aver completato la fase di calcolo in cui moltiplica le cifre a due a due, dovrà sommare i due prodotti parziali. Carlo inizia sommando 7 e 0. Ora deve sommare 1 e 7 e otterrà 8.

Notate che Carlo ha spostato due volte la sua attenzione sulle cifre 1 e 7, ma la prima volta le ha *moltiplicate* e la seconda le ha *sommate* perché il suo "stato mentale" è cambiato.



Inizialmente il nastro conterrà solo i due numeri che vogliamo moltiplicare. Indichiamo con una freccia $\Downarrow$ in quali caselle è focalizzata l'“attenzione” di Carlo, un po' come quando si usa un dito per tenere il segno. Man mano che Carlo calcola, la sua attenzione si sposterà di casella in casella: in alcune caselle scriverà dei simboli e a volte ne cancellerà qualcuno, così da lasciare spazio ad altri simboli. Carlo sa che deve moltiplicare 1×7, per cui ottiene un 7 e lo scrive sul nastro. Ora sposta l'attenzione sulle cifre 3 e 7, che a loro volta vengono moltiplicate.
Dopo aver completato la fase di calcolo in cui moltiplica le cifre a due a due, dovrà sommare i due prodotti parziali. Carlo inizia sommando 7 e 0. Ora deve sommare 1 e 7 e otterrà 8.
Notate che Carlo ha spostato due volte la sua attenzione sulle cifre 1 e 7, ma la prima volta le ha *moltiplicate* e la seconda le ha *sommate* perché il suo “stato mentale” è cambiato.

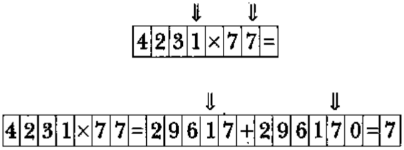


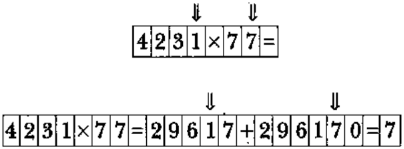
Inizialmente il nastro conterrà solo i due numeri che vogliamo moltiplicare.
Indichiamo con una freccia $\Downarrow$ in quali caselle è focalizzata l'"attenzione" di Carlo, un po' come quando si usa un dito per tenere il segno. Man mano che Carlo calcola, la sua attenzione si sposterà di casella in casella: in alcune caselle scriverà dei simboli e a volte ne cancellerà qualcuno, così da lasciare spazio ad altri simboli.
Carlo sa che deve moltiplicare 1×7, per cui ottiene un 7 e lo scrive sul nastro. Ora sposta l'attenzione sulle cifre 3 e 7, che a loro volta vengono moltiplicate.
Dopo aver completato la fase di calcolo in cui moltiplica le cifre a due a due, dovrà sommare i due prodotti parziali. Carlo inizia sommando 7 e 0. Ora deve sommare 1 e 7 e otterrà 8.
Notate che Carlo ha spostato due volte la sua attenzione sulle cifre 1 e 7, ma la prima volta le ha *moltiplicate* e la seconda le ha *sommate* perché il suo "stato mentale" è cambiato.

$$\begin{array}{r} 4231\times \\ 77 \\ \hline 29617 \\ 29617 \\ \hline 325787 \end{array}$$

											$\Downarrow$							$\Downarrow$			
4	2	3	1	×	7	7	=	2	9	6	1	7	+	2	9	6	1	7	0	=	7

Inizialmente il nastro conterrà solo i due numeri che vogliamo moltiplicare.
Indichiamo con una freccia $\Downarrow$ in quali caselle è focalizzata l'"attenzione" di Carlo, un po' come quando si usa un dito per tenere il segno. Man mano che Carlo calcola, la sua attenzione si sposterà di casella in casella: in alcune caselle scriverà dei simboli e a volte ne cancellerà qualcuno, così da lasciare spazio ad altri simboli.
Carlo sa che deve moltiplicare 1×7, per cui ottiene un 7 e lo scrive sul nastro. Ora sposta l'attenzione sulle cifre 3 e 7, che a loro volta vengono moltiplicate.
Dopo aver completato la fase di calcolo in cui moltiplica le cifre a due a due, dovrà sommare i due prodotti parziali. Carlo inizia sommando 7 e 0. Ora deve sommare 1 e 7 e otterrà 8.
Notate che Carlo ha spostato due volte la sua attenzione sulle cifre 1 e 7, ma la prima volta le ha *moltiplicate* e la seconda le ha *sommate* perché il suo "stato mentale" è cambiato.

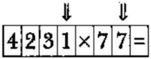




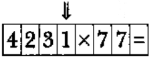
Seconda osservazione: gli stati mentali

Turing capisce che le azioni che Carlo compie sul nastro dipendono non solo dai simboli che sta leggendo, ma anche dal suo stato mentale.

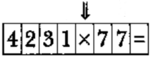
Ma su quante caselle Carlo può simultaneamente posare la sua attenzione? Ne basta *una*!



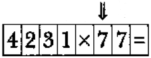
Ma su quante caselle Carlo può simultaneamente posare la sua attenzione? Ne basta *una*!



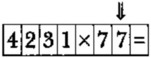
Ma su quante caselle Carlo può simultaneamente posare la sua attenzione? Ne basta *una*!

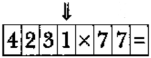


Ma su quante caselle Carlo può simultaneamente posare la sua attenzione? Ne basta *una*!



Ma su quante caselle Carlo può simultaneamente posare la sua attenzione? Ne basta *una*!





Terza osservazione: le testina di lettura

Turing capisce che nulla di essenziale viene perso se assumiamo che Carlo legga un simbolo per volta e che la sua "attenzione" si sposti di una casella alla volta. Il simbolo $\Downarrow$ è anche chiamato testina di lettura.

Abbiamo visto che le azioni che compie Carlo sul nastro consistono nello *scrivere* (oppure nel *cancellare*) un simbolo nella casella su cui si trova la sua attenzione ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra. Abbiamo detto che queste azioni sono determinate dal *simbolo letto* e dal suo *stato mentale*. Ma chi dice a Carlo di volta in volta che azione compiere? Serve una *lista finita di istruzioni*, una "ricetta" per calcolare la moltiplicazione.

Esempio di istruzione

Se stai leggendo il numero 7 e ti trovi nello stato mentale R, sostituisci il simbolo 7 con 3, spostati di una casella a destra e passa nello stato mentale S.

Perché non lo accorciamo un po'? D'ora in poi lo scriviamo simbolicamente come:

$$R\ 7 : 3 \rightarrow S$$

ma se vogliamo andare a *sinistra* oppure *rimanere nella stessa casella* come possiamo scrivere?

$$R\ 7 : 3 \leftarrow S \qquad R\ 7 : 3 \star S$$

Abbiamo visto che le azioni che compie Carlo sul nastro consistono nello *scrivere* (oppure nel *cancellare*) un simbolo nella casella su cui si trova la sua attenzione ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra. Abbiamo detto che queste azioni sono determinate dal *simbolo letto* e dal suo *stato mentale*. Ma chi dice a Carlo di volta in volta che azione compiere? Serve una *lista finita di istruzioni*, una "ricetta" per calcolare la moltiplicazione.

Esempio di istruzione

Se stai leggendo il numero 7 e ti trovi nello stato mentale R, sostituisci il simbolo 7 con 3, spostati di una casella a destra e passa nello stato mentale S.

Perché non lo accorciamo un po'? D'ora in poi lo scriviamo simbolicamente come:

$$R\ 7 : 3 \to S$$

ma se vogliamo andare a *sinistra* oppure *rimanere nella stessa casella* come possiamo scrivere?

$$R\ 7 : 3 \leftarrow S \qquad R\ 7 : 3 \star S$$

Abbiamo visto che le azioni che compie Carlo sul nastro consistono nello *scrivere* (oppure nel *cancellare*) un simbolo nella casella su cui si trova la sua attenzione ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra. Abbiamo detto che queste azioni sono determinate dal *simbolo letto* e dal suo *stato mentale*. Ma chi dice a Carlo di volta in volta che azione compiere? Serve una *lista finita di istruzioni*, una "ricetta" per calcolare la moltiplicazione.

Esempio di istruzione

Se stai leggendo il numero 7 e ti trovi nello stato mentale R, sostituisci il simbolo 7 con 3, spostati di una casella a destra e passa nello stato mentale S.

Perché non lo accorciamo un po'? D'ora in poi lo scriviamo simbolicamente come:

$$R\ 7 : 3 \rightarrow S$$

ma se vogliamo andare a *sinistra* oppure *rimanere nella stessa casella* come possiamo scrivere?

$$R\ 7 : 3 \leftarrow S \qquad R\ 7 : 3 \star S$$

Abbiamo visto che le azioni che compie Carlo sul nastro consistono nello *scrivere* (oppure nel *cancellare*) un simbolo nella casella su cui si trova la sua attenzione ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra. Abbiamo detto che queste azioni sono determinate dal *simbolo letto* e dal suo *stato mentale*. Ma chi dice a Carlo di volta in volta che azione compiere? Serve una *lista finita di istruzioni*, una "ricetta" per calcolare la moltiplicazione.

Esempio di istruzione

Se stai leggendo il numero 7 e ti trovi nello stato mentale R, sostituisci il simbolo 7 con 3, spostatati di una casella a destra e passa nello stato mentale S.

Perché non lo accorciamo un po'? D'ora in poi lo scriviamo simbolicamente come:

$$R\ 7 : 3 \rightarrow S$$

ma se vogliamo andare a *sinistra* oppure *rimanere nella stessa casella* come possiamo scrivere?

$$R\ 7 : 3 \leftarrow S \qquad R\ 7 : 3 \star S$$

Abbiamo visto che le azioni che compie Carlo sul nastro consistono nello *scrivere* (oppure nel *cancellare*) un simbolo nella casella su cui si trova la sua attenzione ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra. Abbiamo detto che queste azioni sono determinate dal *simbolo letto* e dal suo *stato mentale*. Ma chi dice a Carlo di volta in volta che azione compiere? Serve una *lista finita di istruzioni*, una "ricetta" per calcolare la moltiplicazione.

Esempio di istruzione

Se stai leggendo il numero 7 e ti trovi nello stato mentale R, sostituisci il simbolo 7 con 3, spostatati di una casella a destra e passa nello stato mentale S.

Perché non lo accorciamo un po'? D'ora in poi lo scriviamo simbolicamente come:

$$R\ 7 : 3 \rightarrow S$$

ma se vogliamo andare a *sinistra* oppure *rimanere nella stessa casella* come possiamo scrivere?

$$R\ 7 : 3 \leftarrow S \qquad R\ 7 : 3 \star S$$

Abbiamo visto che le azioni che compie Carlo sul nastro consistono nello *scrivere* (oppure nel *cancellare*) un simbolo nella casella su cui si trova la sua attenzione ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra. Abbiamo detto che queste azioni sono determinate dal *simbolo letto* e dal suo *stato mentale*. Ma chi dice a Carlo di volta in volta che azione compiere? Serve una *lista finita di istruzioni*, una "ricetta" per calcolare la moltiplicazione.

Esempio di istruzione

Se stai leggendo il numero 7 e ti trovi nello stato mentale R, sostituisci il simbolo 7 con 3, spostati di una casella a destra e passa nello stato mentale S.

Perché non lo accorciamo un po'? D'ora in poi lo scriviamo simbolicamente come:

$$R\ 7 : 3 \rightarrow S$$

ma se vogliamo andare a *sinistra* oppure *rimanere nella stessa casella* come possiamo scrivere?

$$R\ 7 : 3 \leftarrow S \qquad R\ 7 : 3 \star S$$

Abbiamo visto che le azioni che compie Carlo sul nastro consistono nello *scrivere* (oppure nel *cancellare*) un simbolo nella casella su cui si trova la sua attenzione ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra. Abbiamo detto che queste azioni sono determinate dal *simbolo letto* e dal suo *stato mentale*. Ma chi dice a Carlo di volta in volta che azione compiere? Serve una *lista finita di istruzioni*, una "ricetta" per calcolare la moltiplicazione.

Esempio di istruzione

Se stai leggendo il numero 7 e ti trovi nello stato mentale R, sostituisci il simbolo 7 con 3, spostati di una casella a destra e passa nello stato mentale S.

Perché non lo accorciamo un po'? D'ora in poi lo scriviamo simbolicamente come:

$$R\ 7 : 3 \rightarrow S$$

ma se vogliamo andare a *sinistra* oppure *rimanere nella stessa casella* come possiamo scrivere?

$$R\ 7 : 3 \leftarrow S \qquad R\ 7 : 3 \star S$$

Quarta osservazione: le istruzioni

Turing capisce che è possibile dare una forma molto semplice all'insieme delle istruzioni che determinano le azioni che Carlo dovrà fare. Una istruzione non è che una *quintupla* di una delle sequenti forme:

$$R\ a : b \rightarrow S \qquad R\ a : b \leftarrow S \qquad R\ a : b \star S$$

dove R è uno *stato mentale*, a è il simbolo che leggo, b è il simbolo che *scrivo*, $\rightarrow$, $\leftarrow$, $\star$ indicano dove mi devo *spostare*, e S è il nuovo *stato mentale* in cui mi troverò terminata questa azione.

La coppia "$R\ a$" è chiamata configurazione. La tripla "$b \rightarrow S$" è chiamata azione.

Quarta osservazione: le istruzioni

Turing capisce che è possibile dare una forma molto semplice all'insieme delle istruzioni che determinano le azioni che Carlo dovrà fare. Una istruzione non è che una *quintupla* di una delle sequenti forme:

$$R\ a : b \rightarrow S \qquad R\ a : b \leftarrow S \qquad R\ a : b \star S$$

dove R è uno *stato mentale*, a è il simbolo che leggo, b è il simbolo che *scrivo*, $\rightarrow$, $\leftarrow$, $\star$ indicano dove mi devo *spostare*, e S è il nuovo *stato mentale* in cui mi troverò terminata questa azione.
La coppia "$R\ a$" è chiamata configurazione. La tripla "$b \rightarrow S$" è chiamata azione.

In conclusione, ogni calcolo può essere intrapreso da Carlo come un processo con le seguenti caratteristiche:

- Viene eseguito scrivendo dei *simboli* nelle caselle di un **nastro** di carta, ogni casella può ospitare al più un simbolo.
- Ad ogni passo, Carlo fa **attenzione** al simbolo scritto in *una* sola di queste caselle, e si trova in un determinato **stato mentale**.
- Le azioni di Carlo sono determinate univocamente da un insieme finito di **istruzioni** di una delle seguenti forme:

$$R\ a : b \rightarrow S \qquad R\ a : b \leftarrow S \qquad R\ a : b \star S$$

Le istruzioni sono quindi tali che l'azione successiva di Carlo dipenderà unicamente dalla **configurazione**, ossia dal simbolo che lui sta osservando e dal suo *stato mentale*, e consisterà nello *scrivere* (o *cancellare*) un simbolo nella *casella* osservata ed eventualmente *spostare l'attenzione* sulla casella immediatamente a destra o a sinistra.

Le conclusioni di Turing

In conclusione, ogni calcolo può essere intrapreso da Carlo come un processo con le seguenti caratteristiche:

- Viene eseguito scrivendo dei *simboli* nelle caselle di un **nastro** di carta, ogni casella può ospitare al più un simbolo.
- Ad ogni passo, Carlo fa **attenzione** al simbolo scritto in *una* sola di queste caselle, e si trova in un determinato **stato mentale**.
- Le azioni di Carlo sono determinate univocamente da un insieme finito di **istruzioni** di una delle seguenti forme:

$$R\ a : b \rightarrow S \qquad R\ a : b \leftarrow S \qquad R\ a : b \star S$$

Le istruzioni sono quindi tali che l'**azione** successiva di Carlo dipenderà unicamente dalla **configurazione**, ossia dal *simbolo* che lui sta osservando e dal suo *stato mentale*, e consisterà nello *scrivere* (o cancellare) un simbolo nella *casella* osservata ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra.

Le conclusioni di Turing

In conclusione, ogni calcolo può essere intrapreso da Carlo come un processo con le seguenti caratteristiche:

- Viene eseguito scrivendo dei *simboli* nelle caselle di un **nastro** di carta, ogni casella può ospitare al più un simbolo.
- Ad ogni passo, Carlo fa **attenzione** al simbolo scritto in *una* sola di queste caselle, e si trova in un determinato **stato mentale**.
- Le azioni di Carlo sono determinate univocamente da un insieme finito di **istruzioni** di una delle seguenti forme:

$$R\ a : b \rightarrow S \qquad R\ a : b \leftarrow S \qquad R\ a : b \star S$$

Le istruzioni sono quindi tali che l'**azione** successiva di Carlo dipenderà unicamente dalla **configurazione**, ossia dal *simbolo* che lui sta osservando e dal suo *stato mentale*, e consisterà nello *scrivere* (o cancellare) un simbolo nella *casella* osservata ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra.

Le conclusioni di Turing

In conclusione, ogni calcolo può essere intrapreso da Carlo come un processo con le seguenti caratteristiche:

- Viene eseguito scrivendo dei *simboli* nelle caselle di un **nastro** di carta, ogni casella può ospitare al più un simbolo.
- Ad ogni passo, Carlo fa **attenzione** al simbolo scritto in *una* sola di queste caselle, e si trova in un determinato **stato mentale**.
- Le azioni di Carlo sono determinate univocamente da un insieme finito di **istruzioni** di una delle seguenti forme:

$$R\ a : b \rightarrow S \qquad R\ a : b \leftarrow S \qquad R\ a : b \star S$$

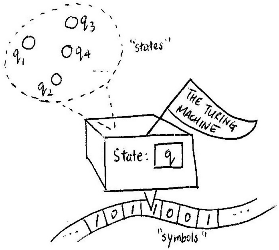
  Le istruzioni sono quindi tali che l'**azione** successiva di Carlo dipenderà unicamente dalla **configurazione**, ossia dal *simbolo* che lui sta osservando e dal suo *stato mentale*, e consisterà nello *scrivere* (o cancellare) un simbolo nella *casella* osservata ed eventualmente *spostare* l'attenzione sulla casella immediatamente a destra o a sinistra.

Quinta osservazione: Carlo ci non serve più!

Turing capisce che il ruolo di Carlo può essere tranquillamente rimpiazzato da una macchina. . . astratta!

Vedremo che questa macchina può essere costruita concretamente: la strisciolina di carta può essere un nastro magnetico, dove i simboli scritti sono rappresentati da informazioni in codice, mentre l'attenzione di Carlo diventa una testina di lettura e gli stati mentali corrispondono a differenti configurazione delle componenti interne della macchina. Le istruzioni della macchina di Turing non sono nient'altro che i programmi.

Per il momento non ci interessa: l'importante è che Turing sia riuscito a dare un *modello matematico* di cosa sia *calcolabile mediante un processo algoritmico*.

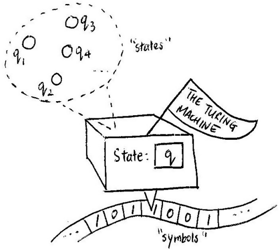


Quinta osservazione: Carlo ci non serve più!

Turing capisce che il ruolo di Carlo può essere tranquillamente rimpiazzato da una macchina. . . astratta!

Vedremo che questa macchina può essere costruita concretamente: la strisciolina di carta può essere un nastro magnetico, dove i simboli scritti sono rappresentati da informazioni in codice, mentre l'attenzione di Carlo diventa una testina di lettura e gli stati mentali corrispondono a differenti configurazione delle componenti interne della macchina. Le istruzioni della macchina di Turing non sono nient'altro che i programmi.

Per il momento non ci interessa: l'importante è che Turing sia riuscito a dare un *modello matematico* di cosa sia *calcolabile mediante un processo algoritmico.*

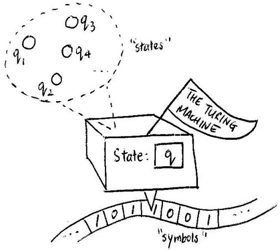


Quinta osservazione: Carlo ci non serve più!

Turing capisce che il ruolo di Carlo può essere tranquillamente rimpiazzato da una macchina. . . astratta!

Vedremo che questa macchina può essere costruita concretamente: la strisciolina di carta può essere un nastro magnetico, dove i simboli scritti sono rappresentati da informazioni in codice, mentre l'attenzione di Carlo diventa una testina di lettura e gli stati mentali corrispondono a differenti configurazione delle componenti interne della macchina. Le istruzioni della macchina di Turing non sono nient'altro che i programmi.

Per il momento non ci interessa: l'importante è che Turing sia riuscito a dare un *modello matematico* di cosa sia *calcolabile mediante un processo algoritmico*.



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \square \rightarrow O$

(ii) $O\ 4 : \square \rightarrow E$

(iii) $O\ 8 : \square \rightarrow E$

(iv) $O\ \square : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \square \rightarrow O$

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \square \rightarrow O$

(ii) $O\ 4 : \square \rightarrow E$

(iii) $O\ 8 : \square \rightarrow E$

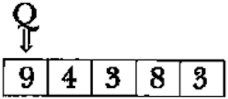
(iv) $O\ \square : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \square \rightarrow O$

Sulla macchina scrivo l'input 94383, ovvero il numero in ingresso. Q è chiamato stato iniziale, con la testina di lettura nella casella più a destra. Mi trovo nella cosiddetta configurazione iniziale.



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \square \rightarrow O$

(ii) $O\ 4 : \square \rightarrow E$

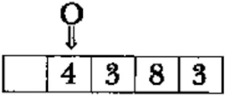
(iii) $O\ 8 : \square \rightarrow E$

(iv) $O\ \square : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \square \rightarrow O$



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \square \rightarrow O$

(ii) $O\ 4 : \square \rightarrow E$

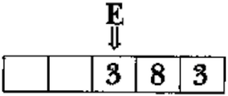
(iii) $O\ 8 : \square \rightarrow E$

(iv) $O\ \square : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \square \rightarrow O$



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \square \rightarrow O$

(ii) $O\ 4 : \square \rightarrow E$

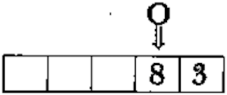
(iii) $O\ 8 : \square \rightarrow E$

(iv) $O\ \square : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \square \rightarrow O$



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \Box \rightarrow O$

(ii) $O\ 4 : \Box \rightarrow E$

(iii) $O\ 8 : \Box \rightarrow E$

(iv) $O\ \Box : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \Box \rightarrow O$

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \Box \to O$

(ii) $O\ 4 : \Box \to E$

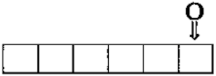
(iii) $O\ 8 : \Box \to E$

(iv) $O\ \Box : 1 \star F$

(v) $Q\ 1 : 1 \to Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \Box \to O$



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \Box \rightarrow O$

(ii) $O\ 4 : \Box \rightarrow E$

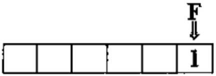
(iii) $O\ 8 : \Box \rightarrow E$

(iv) $O\ \Box : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \Box \rightarrow O$



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \Box \rightarrow O$

(ii) $O\ 4 : \Box \rightarrow E$

(iii) $O\ 8 : \Box \rightarrow E$

(iv) $O\ \Box : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \Box \rightarrow O$

Attenzione!!! Il calcolo *non termina* su input 12:

Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \square \rightarrow O$

(ii) $O\ 4 : \square \rightarrow E$

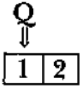
(iii) $O\ 8 : \square \rightarrow E$

(iv) $O\ \square : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \square \rightarrow O$



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \Box \rightarrow O$

(ii) $O\ 4 : \Box \rightarrow E$

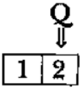
(iii) $O\ 8 : \Box \rightarrow E$

(iv) $O\ \Box : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \Box \rightarrow O$



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \square \rightarrow O$

(ii) $O\ 4 : \square \rightarrow E$

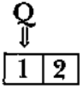
(iii) $O\ 8 : \square \rightarrow E$

(iv) $O\ \square : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \square \rightarrow O$



Un esempio di macchina di Turing

Ogni macchina è definita da un insieme finito e fissato di istruzioni. Nel nostro caso, le istruzioni sono:

(i) $E\ 3 : \Box \rightarrow O$

(ii) $O\ 4 : \Box \rightarrow E$

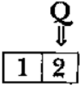
(iii) $O\ 8 : \Box \rightarrow E$

(iv) $O\ \Box : 1 \star F$

(v) $Q\ 1 : 1 \rightarrow Q$

(vi) $Q\ 2 : 2 \leftarrow Q$

(vii) $Q\ 9 : \Box \rightarrow O$



Un esempio di macchina di Turing

Alcune conclusioni

Possiamo affermare che:

- Il comportamento computazionale della macchina dipende dall'*input*.
- Su certi input il calcolo di una macchina *termina*, su altri il calcolo *non termina*.
- La macchina deve essere *deterministica*: non ho mai due istruzioni che "cominciano" nello stesso modo:

$$E\ 3 : \Box \rightarrow O$$

$$E\ 3 : \Box \leftarrow Q$$

Un esempio di macchina di Turing

Alcune conclusioni

Possiamo affermare che:

- Il comportamento computazionale della macchina dipende dall'*input*.
- Su certi input il calcolo di una macchina *termina*, su altri il calcolo *non termina*.
- La macchina deve essere *deterministica*: non ho mai due istruzioni che "cominciano" nello stesso modo:

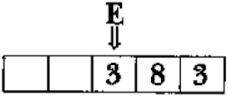
$$E\,3 : \square \to O$$
$$E\,3 : \square \leftarrow Q$$

Alcune conclusioni

Possiamo affermare che:

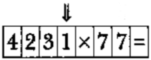
- Il comportamento computazionale della macchina dipende dall'*input*.
- Su certi input il calcolo di una macchina *termina*, su altri il calcolo *non termina*.
- La macchina deve essere *deterministica*: non ho mai due istruzioni che "cominciano" nello stesso modo:

$$E\ 3 : \square \rightarrow O$$
$$E\ 3 : \square \leftarrow Q$$



Macchina di Turing universale e moderni calcolatori

- Ogni macchina di Turing $\mathcal{M}$ ha un insieme di istruzioni finito e fissato. Ad esempio, posso definire una macchina di Turing $\mathcal{P}$ che calcola la moltiplicazione. Per ogni ingresso che inserirò nel nastro, ad esempio 4231×77, l'insieme di istruzioni di questa macchina permetterà di calcolare il risultato della moltiplicazione, ovvero 325787, il quale si troverà scritto nel nastro al termine del calcolo.
- Ma l'insieme delle istruzioni di una macchina di Turing $\mathcal{M}$ è un insieme di simboli, i quali possono a loro volta essere scritti sul nastro. Chiamiamo *codice numerico di* $\mathcal{M}$ la stringa di tutti i simboli di tutte le istruzioni di $\mathcal{M}$.
- Turing riesce a definire una macchina di Turing $\mathcal{U}$ che partendo da un nastro riempito con il codice numerico di $\mathcal{M}$ e da un ingresso di $\mathcal{M}$, riesce a "simulare" il calcolo di $\mathcal{M}$ su quell'ingresso. Ad esempio, se scrivessi sul nastro il codice numerico di $\mathcal{P}$ e 4231×77, la macchina di Turing $\mathcal{U}$ calcolerebbe 325787.
- $\mathcal{U}$ si chiama macchina di Turing universale, ed è il precursore (teorico) dei moderni *interpreti*.



- Ogni macchina di Turing $\mathcal{M}$ ha un insieme di istruzioni finito e fissato. Ad esempio, posso definire una macchina di Turing $\mathcal{P}$ che calcola la moltiplicazione. Per ogni ingresso che inserirò nel nastro, ad esempio 4231×77, l'insieme di istruzioni di questa macchina permetterà di calcolare il risultato della moltiplicazione, ovvero 325787, il quale si troverà scritto nel nastro al termine del calcolo.
- Ma l'insieme delle istruzioni di una macchina di Turing $\mathcal{M}$ è un insieme di simboli, i quali possono a loro volta essere scritti sul nastro. Chiamiamo *codice numerico di* $\mathcal{M}$ la stringa di tutti i simboli di tutte le istruzioni di $\mathcal{M}$.
- Turing riesce a definire una macchina di Turing $\mathcal{U}$ che partendo da un nastro riempito con il codice numerico di $\mathcal{M}$ e da un ingresso di $\mathcal{M}$, riesce a "simulare" il calcolo di $\mathcal{M}$ su quell'ingresso. Ad esempio, se scrivessi sul nastro il codice numerico di $\mathcal{P}$ e 4231×77, la macchina di Turing $\mathcal{U}$ calcolerebbe 325787.
- $\mathcal{U}$ si chiama macchina di Turing universale, ed è il precursore (teorico) dei moderni *interpreti*.

- Ogni macchina di Turing $\mathcal{M}$ ha un insieme di istruzioni finito e fissato. Ad esempio, posso definire una macchina di Turing $\mathcal{P}$ che calcola la moltiplicazione. Per ogni ingresso che inserirò nel nastro, ad esempio 4231×77, l'insieme di istruzioni di questa macchina permetterà di calcolare il risultato della moltiplicazione, ovvero 325787, il quale si troverà scritto nel nastro al termine del calcolo.
- Ma l'insieme delle istruzioni di una macchina di Turing $\mathcal{M}$ è un insieme di simboli, i quali possono a loro volta essere scritti sul nastro. Chiamiamo *codice numerico di* $\mathcal{M}$ la stringa di tutti i simboli di tutte le istruzioni di $\mathcal{M}$.
- Turing riesce a definire una macchina di Turing $\mathcal{U}$ che partendo da un nastro riempito con il codice numerico di $\mathcal{M}$ e da un ingresso di $\mathcal{M}$, riesce a "simulare" il calcolo di $\mathcal{M}$ su quell'ingresso. Ad esempio, se scrivessi sul nastro il codice numerico di $\mathcal{P}$ e 4231×77, la macchina di Turing $\mathcal{U}$ calcolerebbe 325787.
- $\mathcal{U}$ si chiama macchina di Turing universale, ed è il precursore (teorico) dei moderni *interpreti*.

Codice numerico di una macchina di Turing $\mathcal{M}$	Ingresso di $\mathcal{M}$

- Ogni macchina di Turing $\mathcal{M}$ ha un insieme di istruzioni finito e fissato. Ad esempio, posso definire una macchina di Turing $\mathcal{P}$ che calcola la moltiplicazione. Per ogni ingresso che inserirò nel nastro, ad esempio 4231×77, l'insieme di istruzioni di questa macchina permetterà di calcolare il risultato della moltiplicazione, ovvero 325787, il quale si troverà scritto nel nastro al termine del calcolo.
- Ma l'insieme delle istruzioni di una macchina di Turing $\mathcal{M}$ è un insieme di simboli, i quali possono a loro volta essere scritti sul nastro. Chiamiamo *codice numerico di* $\mathcal{M}$ la stringa di tutti i simboli di tutte le istruzioni di $\mathcal{M}$.
- Turing riesce a definire una macchina di Turing $\mathcal{U}$ che partendo da un nastro riempito con il codice numerico di $\mathcal{M}$ e da un ingresso di $\mathcal{M}$, riesce a "simulare" il calcolo di $\mathcal{M}$ su quell'ingresso. Ad esempio, se scrivessi sul nastro il codice numerico di $\mathcal{P}$ e 4231×77, la macchina di Turing $\mathcal{U}$ calcolerebbe 325787.
- $\mathcal{U}$ si chiama macchina di Turing universale, ed è il precursore (teorico) dei moderni *interpreti*.

Il calcolatore universale è un apparato concettuale meraviglioso, capace di eseguire da solo qualsiasi compito di natura algoritmica: ma è materialmente possibile costruirlo? Avrebbe saputo risolvere problemi del mondo reale in un tempo ragionevole con una quantità ragionevole di risorse? Turing non si limita a *pensare* a questa possibilità ... e inizia a rimboccarsi le maniche!

- Nel primo dopoguerra, Von Neumann lavora negli Stati Uniti al calcolatore *ENIAC* (*Electronic Numerical Integrator and Computer*), e successivamente al calcolatore *EDVAC* (*Electronic Discrete Variable Automatic Calculator*).
- Negli stessi anni, Turing lavora nel Regno Unito al calcolatore *ACE* (*Automatic Computing Engine*).

Sono i primissimi computer elettronici digitali della storia!

Il calcolatore universale è un apparato concettuale meraviglioso, capace di eseguire da solo qualsiasi compito di natura algoritmica: ma è materialmente possibile costruirlo? Avrebbe saputo risolvere problemi del mondo reale in un tempo ragionevole con una quantità ragionevole di risorse? Turing non si limita a *pensare* a questa possibilità ...e inizia a rimboccarsi le maniche!

- Nel primo dopoguerra, Von Neumann lavora negli Stati Uniti al calcolatore *ENIAC* (*Electronic Numerical Integrator and Computer*), e successivamente al calcolatore *EDVAC* (*Electronic Discrete Variable Automatic Calculator*).
- Negli stessi anni, Turing lavora nel Regno Unito al calcolatore *ACE* (*Automatic Computing Engine*).

Sono i primissimi computer elettronici digitali della storia!

Il calcolatore universale è un apparato concettuale meraviglioso, capace di eseguire da solo qualsiasi compito di natura algoritmica: ma è materialmente possibile costruirlo? Avrebbe saputo risolvere problemi del mondo reale in un tempo ragionevole con una quantità ragionevole di risorse? Turing non si limita a *pensare* a questa possibilità . . . e inizia a rimboccarsi le maniche!

- Nel primo dopoguerra, Von Neumann lavora negli Stati Uniti al calcolatore *ENIAC* (*Electronic Numerical Integrator and Computer*), e successivamente al calcolatore *EDVAC* (*Electronic Discrete Variable Automatic Calculator*).
- Negli stessi anni, Turing lavora nel Regno Unito al calcolatore *ACE* (*Automatic Computing Engine*).

Sono i primissimi computer elettronici digitali della storia!

Il calcolatore universale è un apparato concettuale meraviglioso, capace di eseguire da solo qualsiasi compito di natura algoritmica: ma è materialmente possibile costruirlo? Avrebbe saputo risolvere problemi del mondo reale in un tempo ragionevole con una quantità ragionevole di risorse? Turing non si limita a *pensare* a questa possibilità ... e inizia a rimboccarsi le maniche!

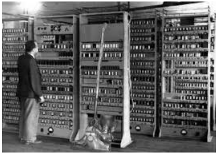
- Nel primo dopoguerra, Von Neumann lavora negli Stati Uniti al calcolatore *ENIAC* (*Electronic Numerical Integrator and Computer*), e successivamente al calcolatore *EDVAC* (*Electronic Discrete Variable Automatic Calculator*).
- Negli stessi anni, Turing lavora nel Regno Unito al calcolatore *ACE* (*Automatic Computing Engine*).

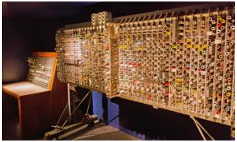
Sono i primissimi computer elettronici digitali della storia!

Il calcolatore universale è un apparato concettuale meraviglioso, capace di eseguire da solo qualsiasi compito di natura algoritmica: ma è materialmente possibile costruirlo? Avrebbe saputo risolvere problemi del mondo reale in un tempo ragionevole con una quantità ragionevole di risorse? Turing non si limita a *pensare* a questa possibilità ... e inizia a rimboccarsi le maniche!

- Nel primo dopoguerra, Von Neumann lavora negli Stati Uniti al calcolatore *ENIAC* (*Electronic Numerical Integrator and Computer*), e successivamente al calcolatore *EDVAC* (*Electronic Discrete Variable Automatic Calculator*).
- Negli stessi anni, Turing lavora nel Regno Unito al calcolatore *ACE* (*Automatic Computing Engine*).

Sono i primissimi computer elettronici digitali della storia!





Grazie per l'attenzione!

https://www.youtube.com/watch?v=FTSAiF9AHN4
https://www.youtube.com/watch?v=vo8izCKHiF0