L’informazione da elaborare è scritta su un nastro di lunghezza indefinita con caratteri di un alfabeto finito di simboli digitali;

l’elaborazione è suddivisa in passi elementari e ad ogni passo viene preso in esame uno di questi caratteri;

le azioni da eseguire nell’intervallo i-esimo sono definite da una istruzione descritta da una **quintupla** di elementi:

* 1. **Si**: sigla che identifica l’istruzione corrente;
  2. **Xi**: il simbolo letto sul nastro;
  3. **Yi**: il simbolo che deve sostituire, eventualmente, **Xi**
  4. **Vi**: spostamento (destra/sinistra) per il prossimo carattere,
  5. **Si+1**: sigla che identifica l’istruzione successiva.

Questo strumento metodologico, concettualmente semplice, è stato etichettato come “**macchina di Turing”;** con questo dispositivo è possibile risolvere una classe di problemi che, secondo una celebre congettura di Alonzo Church, coincide con quella delle funzioni computabili.

Turing dimostrò che non è risolubile il problema dell'arresto: in altri termini non è possibile con una macchina di Turing decidere se un'altra macchina di Turing si arresterà o no, dati un programma e un nastro iniziale.

Turing ha poi mostrato che è possibile una **macchina universale**, cioè una macchina di Turing in grado di *imitare* una qualsiasi particolare macchina di Turing.

La macchina universale di Turing ha costituito modello astratto dei futuri linguaggi di programmazione.

Turing ha proposto un linguaggio per descrivere manipolazioni di simboli digitali.

I simboli da manipolare, i dati del problema, sono scritti su un nastro

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

Ad ogni passo della manipolazione viene preso in considerazione un carattere. Il programma che definisce la manipolazione è un elenco di istruzioni descrivibili come quintuple del tipo seguente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sigla/Indirizzo istruzione  corrente | Contenuto  corrente sul nastro | Nuovo  contenuto | Spostamento per la prossima lettura | Sigla/Indirizzo prossima  istruzione |

Esempio di programma che sostituisce una cifra con 0 se la cifra è pari o con 1 se la cifra è dispari.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | 1 | 1 | = | Z |
| A | 2 | 0 | = | Z |
| A | 3 | 1 | = | Z |
| A | 4 | 0 | = | Z |
| A | 5 | 1 | = | Z |
| A | 6 | 0 | = | Z |
| A | 7 | 1 | = | Z |
| A | 8 | 0 | = | Z |
| A | 9 | 1 | = | Z |
| A | 0 | 0 | = | Z |
| Z | 0 | 0 | = | Z |
| Z | 1 | 1 | = | Z |

Qualunque sia lo stato iniziale del nastro, sarà eseguita una azione di tipo A e una di tipo Z e alla fine il nastro conterrà la cifra 0 se la cifra iniziale è pari o la cifra 1 se dispari. Esempio di output.

|  |  |
| --- | --- |
| input | output |
| |  |  |  | | --- | --- | --- | | Θ | 5 | Θ | | |  |  |  | | --- | --- | --- | | Θ | 1 | Θ | |
| |  |  |  | | --- | --- | --- | | Θ | 8 | Θ | | |  |  |  | | --- | --- | --- | | Θ | 0 | Θ | |

Altro esempio

Scrivere il programma che cancella il numero corrente e scrive sulla casella successiva 1 se il numero letto è dispari e 0 se è pari (vedi tabella)

|  |  |
| --- | --- |
| input | output |
| |  |  |  | | --- | --- | --- | | Θ | 5 | Θ | | |  |  |  | | --- | --- | --- | | Θ | Θ | 1 | |
| |  |  |  | | --- | --- | --- | | Θ | 8 | Θ | | |  |  |  | | --- | --- | --- | | Θ | Θ | 0 | |

Il programma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | 1 | Θ | > | D |
| A | 2 | Θ | > | P |
| A | 3 | Θ | > | D |
| A | 4 | Θ | > | P |
| A | 5 | Θ | > | D |
| A | 6 | Θ | > | P |
| A | 7 | Θ | > | D |
| A | 8 | Θ | > | P |
| A | 9 | Θ | > | D |
| A | 0 | Θ | > | P |
| P | Θ | 0 | < | Z |
| D | Θ | 1 | < | Z |
| Z | Θ | Θ | = | Z |

Esempio di esecuzione del programma

|  |  |  |
| --- | --- | --- |
| nastro | stato | nastro |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | .. | **5** | Θ | .. | .. | | .. | Θ | **Θ** | .. | .. | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | A | **5** | Θ | > | D | | D | **Θ** | 1 | < | Z | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | .. | Θ | **Θ** | .. | .. | | .. | **Θ** | 1 | .. | .. | |

|  |  |  |
| --- | --- | --- |
| nastro | stato | nastro |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | .. | **4** | Θ | .. | .. | | .. | Θ | **Θ** | .. | .. | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | A | **4** | Θ | > | P | | P | **Θ** | 0 | < | Z | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | .. | Θ | **Θ** | .. | .. | | .. | **Θ** | 0 | .. | .. | |

In entrambi i casi la manipolazione ha termine nello stato Z

|  |  |  |
| --- | --- | --- |
| nastro | stato | nastro |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | .. | **Θ** | 1 | .. | .. | | .. | **Θ** | 0 | .. | .. | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Z | **Θ** | Θ | = | Z | | Z | **Θ** | Θ | = | Z | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | .. | **Θ** | 1 | .. | .. | | .. | **Θ** | 0 | .. | .. | |