**IL CALCOLO DI BOOLE, FREGE E TURING**

**La proposta di Boole**

Fallito il tentativo (Hobbes) di disciplinare il linguaggio naturale al fine di utilizzarlo per descrivere argomentazioni cogenti, Leibniz propone la creazione di un linguaggio formale.

Alcuni esempi di “calcolo filosofico” proposto da Leibniz

Se A e B sono due termini allora AΘB è un termine e AΘA = A.

Per Boole gli oggetti del discorso sono le classi:

* la classe degli uomini, la classe dei viventi, la classe dei mammiferi, la classe delle cose bianche …
* le regole o le operazioni possibili tra i simboli che rappresentano queste classi sono quelle dell’algebra.

Le classi si possono rappresentare con simboli e questi simboli possono essere usati in operazioni tipo algebra (come per la somma e la differenza proposta da Hobbes).

Cosa sono le operazioni + , - , × in questo caso?

Come vanno interpretate?

**L’algebra delle classi secondo Boole**

Se X rappresenta la classe delle cose bianche, Y quella degli animali e Z quella dei cornuti allora

X×Y è la classe degli animali bianchi e

X×Y×Z è la classe degli animali bianchi cornuti

L’operazione × è l’intersezione di due classi

**teorema X×X = X2 = X**

X×X = X solo per X = 0 e X = 1

0×X = 0 e 1×X = X

In questa algebra 0 è la classe vuota e 1 è la classe della totalità delle cose. Inoltre

X + Y rappresenta l’unione delle due classi

X – Y è la classe degli oggetti che sono in X, ma non in Y:

1 – X è la classe di tutti gli oggetti che non sono in X

X + (1 – X ) = 1 I conti tornano!

Primo esempio di calcolo

X×X = X 🡺 X2 = X 🡺 X - X2 = 0 🡺 **X×(1 – X) = 0**

Esprime che niente può appartenere e non appartenere a una classe: *tertium non datur*!

è il principio di non contraddizione di Aristotele ottenuto da un calcolo!

**Il primo esempio del *calculemus* di Leibniz!**

Seguendo il progetto del *calculemus*  di Leibniz, Boole propone un linguaggio artificiale, un’algebra con due operazioni:

**l’algebra della logica**

0 + 0 = 0 0 \* 0 = 0 X + X = X X\*X = X

0 + 1 = 1 0 \* 1 = 0 X + 1 = 1 X\*1 = X

1 + 0 = 1 1 \* 0 = 0 1 + X = 1 1\*X = X

1 + 1 = 1 1 \* 1 = 1 X + X = X X\*X = X

Le proprietà distributiva, commutativa e associativa

A + B = B + A; A\*B = B\*A

(A + B) + C = A + (B + C);

(A \* B) \* C = A \* (B \* C)

(A+B)\*C = A\*C+B\*C

E’ un principio generale del linguaggio (non solo di quello matematico) che sia consentito di usare simboli (come nomi) per rappresentare qualunque cosa si scelga di voler rappresentare. (Vedi Aristotele e Leibniz)

---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Uso di simboli per rappresentare classi o insiemi**

0 è vuoto, 1 è universo; + è l’unione, \* è intersezione.

Il dispositivo linguistico di Boole descrive **unione e intersezione di insiemi**

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Uso di simboli per rappresentare proposizioni**

0 è falso, 1 è vero; + è OR e × è AND

Il dispositivo linguistico di Boole descrive **il calcolo delle proposizioni**

**Il linguaggio di Boole consente di descrivere non solo l’unione e la intersezione tra insiemi, ma anche il calcolo delle proposizioni:**

**è sufficiente interpretare**

**0 come falso e 1 come vero;**

**+ come l’operatore ꓦ**

**× come l’operatore ꓥ**

**La proposta di Frege**

Il sistema di Boole dimostra come una deduzione può essere trattata come un calcolo. Tuttavia questo sistema è ancora limitato. Per esempio con questa algebra non si riesce a formulare la seguente affermazione

**Tutti gli studenti bocciati sono pigri o disinteressati.**

Con l’algebra di Boole non è possibile distinguere la classe dei pigri da quella dei disinteressati.

Frege propone un linguaggio (di programmazione?) seguendo l’idea di Leibniz di lingua universale la cui potenza espressiva dipende dalla scelta dei simboli usati.

In Boole le proposizioni sono costanti con valore 0 o 1 non possono contenere variabili che ne determino il vapore

*Begriffsschrift* o Ideografia, la lingua di Frege.

**Tutti i cavalli sono mammiferi** diventa

**se x è un cavallo 🡺 x è un mammifero**

(ꓯx)(cavallo(x) 🡺 mammifero(x))

**Alcuni cavalli sono purosangue** diventa

**x è un cavallo e x è un purosangue**

(ꓱx)(cavallo(x) ꓥ purosangue(x))

**Tutti gli studenti bocciati sono pigri o disinteressati.**

(ꓯx)(bocciato(x) 🡺 pigro(x) ꓦ disinteressato(x))

Da questi esempi si vede che Frege non sta solo elaborando un trattamento matematico della logica, ma di fatto sta creando **un nuovo linguaggio artificiale** (Begriffsschrift) col quale scrivere inferenze logiche come operazioni eseguibili in modo meccanico su simboli (cioè senza la necessità di capirne il significato).

Il linguaggio di Frege è un ulteriore passo avanti rispetto al progetto di Leibniz: consente, in linea di principio, di descrivere ragionamenti (matematici), ma non garantisce di raggiungere sempre un risultato concreto in un tempo determinato.

**Quo facto, definito un linguaggio formale, calculemus!**

Utilizzando questo linguaggio, Frege e Russell propongono un sistema formale per la logica in analogia a quello di Euclide per la geometria.

* Russell aggiunge al sistema formale della logica assiomi per dedurre proposizioni matematiche con l’intento di derivare tutte le proposizioni dell’aritmetica.
* La macchina per l’aritmetica di Russell suscita molte perplessità (vedi Poincaré).
* Per dimostrare che il sistema proposto da Russell è completo e non contradditorio, Hilbert propone due problemi.

1.Gödel dimostra che il sistema proposto da Frege Russell per la logica è completo: dagli assiomi e dalle regole di inferenza sono deducibili tutte e sole le proposizioni vere.

2.Gödel dimostra che in ogni formalizzazione [coerente](https://it.wikipedia.org/wiki/Coerenza_(logica_matematica)) della matematica (che sia sufficientemente potente da poter [assiomatizzare](https://it.wikipedia.org/wiki/Assioma_(matematica)) la teoria elementare dei [numeri naturali](https://it.wikipedia.org/wiki/Numeri_naturali)) è possibile costruire una proposizione sintatticamente corretta che non può essere né dimostrata né confutata all'interno dello stesso sistema. Quindi il problema della decisione (***Enscheidungsproblem***) non è risolubile con un algoritmo.

**La proposta di Turing**

Turing nel 1936 pubblica l’articolo

*On Computable Numbers with an Application to the Entscheidungsproblem*

In questo articolo viene presentato

* un linguaggio effettivo (*macchina virtuale)* per manipolare simboli digitali con la **potenzialità espressiva del calcolabile** e
* un esempio di problema non calcolabile: quindi si dimostra che il secondo problema di Hilbert non è risolubile.

**Per eseguire una particolare elaborazione si deve**

* Riportare su nastro i caratteri che identificano i dati iniziali
* Predisporre l’elenco dei singoli stati che definiscono il tipo di manipolazione sul simbolo corrente
* Individuare lo stato iniziale e
* Individuare la posizione del primo carattere da trattare.

**Se la descrizione del procedimento è corretta, l’elaborazione termina in un tempo finito e lasciando sul nastro la descrizione del risultato**.

**Ogni procedimento di soluzione descritto col metodo di Turing si dice macchina di Turing**

**Una macchina di Turing predisposta per risolvere un particolare problema è, di fatto, un programma/algoritmo.**

Turing ha proposto una macchina particolare (la MUT) che sa eseguire l’elaborazione descritta da una qualsiasi macchina di Turing riportata sul nastro.

**La macchina universale di Turing è, di fatto, un interprete.**

I simboli da manipolare sono disposti su un nastro

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .. | .. | 1 | 7 | 2 | 4 | 3 | .. | .. |

Ogni istruzione agisce su uno dei caratteri scritti sul nastro prendendone in considerazione uno alla volta; sono possibili tre tipi di azioni:

* azione sul carattere corrente, per eventuale sostituzione con altro carattere
* scelta dello spostamento di un posto a destra o a sinistra ,
* scelta della istruzione da eseguire come successiva

Le istruzioni hanno la seguente struttura

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Indirizzo istruzione**  **corrente** | **Carattere**  **corrente** | **Nuovo**  **carattere** | **Direzione dello**  **spostamento** | **Indirizzo istruzione**  **successiva** |

Le singole manipolazioni sono descritte da un programma rappresentato da un elenco di istruzioni; l’indirizzo delle istruzioni è rappresentato da una sigla, il carattere “blank” è rappresentato da Θ e gli spostamenti per individuare il carattere da esaminare nella istruzione successiva sono <, per il carattere di sinistra, > per quello di destra e = per rimanere sulla stessa posizione.

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 | .. | .. | |

Programma per verificare se un numero intero (di n cifre) è pari o dispari; il programma deve cancellare il numero dato e aggiungere alla sua destra la cifra 0 se il numero è pari e 1 se è dispari;

se il numero da verificare è il seguente

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **5** | 7 | 2 | 4 | 3 | Θ |  |

L’output deve essere

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Θ | Θ | Θ | Θ | Θ | 1 |  |

Si suppone di far partire l’indagine dalla cifra più a sinistra; a destra è necessario il simbolo Θ per segnalare il termine del numero oggetto dell’indagine.

I passi intermedi della elaborazione sono i seguenti

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| … | …. | **5** | 7 | 2 | 4 | 3 | Θ | …. | …. |
| … | …. | Θ | **7** | 2 | 4 | 3 | Θ | …. | …. |
| … | …. | Θ | Θ | **2** | 4 | 3 | Θ | …. | …. |
| … | …. | Θ | Θ | Θ | **4** | 3 | Θ | …. | …. |
| … | …. | Θ | Θ | Θ | Θ | **3** | Θ | …. | …. |
| … | …. | Θ | Θ | Θ | Θ | Θ | **Θ** | …. | …. |
| … | …. | Θ | Θ | Θ | Θ | Θ | **1** | …. | …. |
| … | … | Θ | Θ | Θ | Θ | **Θ** | 1 | …. | …. |

L’insieme degli stati che costituiscono il programma, cioè l’elenco degli stati che descrivono la manipolazione richiesta per ogni possibile contenuto iniziale lecito del nastro, è riportato nella seguente tabella.

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