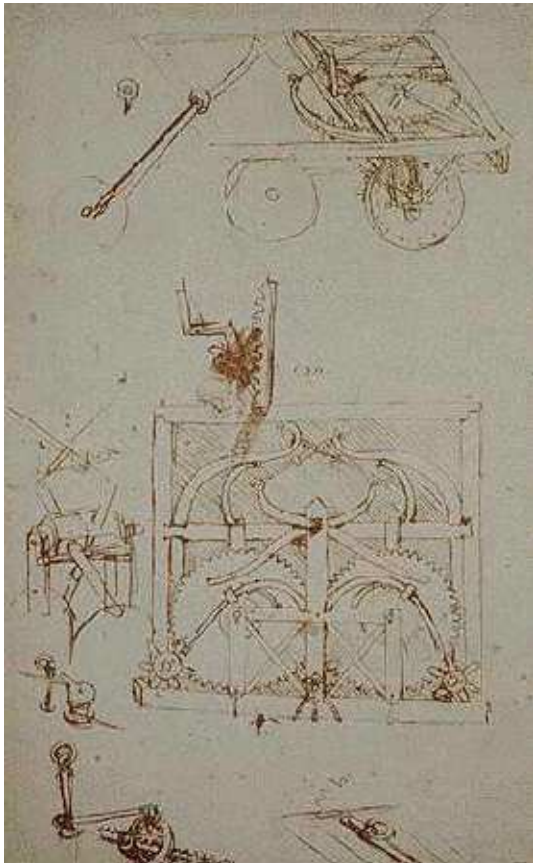


## CAPITOLO 2.

### MACCHINE A MEMORIA FINITA: GLI AUTOMI



**Leonardo da Vinci, 1478,  
(foglio 812).**

Per alcuni il disegno rappresenta il progetto di una sorta di automobile. Per altri (ad es. l'esperto di robotica Mark Elling Rosheim) è invece un disegno di "automa programmabile" capace di compiere un percorso ben preciso fermandosi e ruotando a destra o a sinistra secondo un programma prestabilito.

#### **1. Macchine che ricordano: gli automi**

In questo capitolo vogliamo dare una definizione matematica di "macchina" con l'intenzione di comprendere nella nostra definizione sia i calcolatori elettronici che usiamo tutti i giorni sia le macchinette distributrici di bevande, sia gli ascensori. Per fare questo, faremo astrazione :

- dai materiali di cui possono essere costituiti tali marchingegni,
- dai costi di produzione,
- dal consumo di energia,
- dai tempi di risposta
- dal modo come sono costruiti
- dal problema della efficienza

e così via<sup>1</sup>. Inoltre perverremo alla nostra definizione con una analisi del tipo "a scatola nera". Ciò significa che nel considerare una macchina reale si rinuncia ad ogni indagine sulla sua struttura interna o sui principi che ne regolano il funzionamento. Ci si limita solo ad esaminare il suo "comportamento" descrivendo le risposte che essa fornisce (output) alla successione di stimoli ricevuti (input). Cominciamo ad esaminare alcuni esempi.

**L'ascensore senza memoria.** Supponiamo di essere in un ascensore senza gettoniera e di volere descriverne il comportamento. Come tutte le macchine, l'ascensore è pronto a ricevere degli stimoli, che possiamo rappresentare con l'insieme  $X = \{0,1,2,A\}$  dove:

- 0,1,2 indicano i tre pulsanti con cui si chiede di andare al piano 0, 1 e 2 rispettivamente
- A è il pulsante dell'allarme.

Le possibili risposte a tali stimoli possono inoltre essere rappresentate dall'insieme  $Y = \{P_0, P_1, P_2, S\}$  i cui elementi indicano rispettivamente le azioni di:

- andare al piano terra, primo piano, secondo piano
- suonare l'allarme.

In tale caso il comportamento della macchina è descrivibile dalla semplice funzione  $f: X \rightarrow Y$  che associa ad ogni comando l'azione voluta. La tabella di tale funzione è la seguente:

$$f(0) = P_0, f(1) = P_1, f(2) = P_2, f(A) = S.$$

Semberebbe da tale esempio che il concetto di funzione sia adeguato a rappresentare il comportamento di una macchina. Una macchina può però essere un poco più intelligente, ad esempio potrebbe ricordare alcune delle cose accadute precedentemente e quindi agire non solo in base all' input ricevuto in un dato istante ma anche in base a ciò che "ricorda" degli input precedenti.

**Gli ascensori con memoria.** Supponiamo ad esempio che l'ascensore abbia una gettoniera, allora bisogna aggiungere ai possibili input una lettera, ad esempio la lettera  $M$ , che indichi l'inserimento di una moneta. Inoltre tra i possibili output deve essercene anche uno, chiamiamolo  $N$ , che significhi "non fare nulla". Infatti se viene chiesto all'ascensore di andare ad un certo piano senza aver messo prima una moneta, allora l'ascensore deve "avere l'intelligenza" di non muoversi. A questo punto il concetto di funzione non è più sufficiente perché agli stimoli 1, 2, 3 possono corrispondere sia le risposte  $P_1, P_2, P_3$  che la risposta  $N$ .

*La risposta dipende non solo dallo stimolo ad un certo istante ma anche dagli stimoli che sono stati forniti in precedenza.*

Indicheremo con  $P$  la memorizzazione di "pagato" e con  $NP$  quella di "non pagato" e tali possibili memorizzazioni devono essere viste come "stati" che la macchina può assumere in conseguenza degli stimoli ricevuti. Pertanto:

- in ogni istante la macchina è in uno *stato*, stato che ha raggiunto in base a stimoli precedenti
- la risposta ad un input dipende sia dall'input ricevuto sia dallo stato in cui si trova la macchina;
- gli input determinano non solo gli output ma anche il passaggio da uno stato ad un altro.

<sup>1</sup> D'altra parte è una caratteristica della matematica quella di "fare astrazioni" in modo che l'oggetto di investigazione sia abbastanza semplice da potere essere studiato. Una cosa analoga avviene in geometria dove, per esempio, non si considerano i colori, gli odori, i pesi degli oggetti da esaminare. La stessa nozione di punto è il risultato di un processo di astrazione per cui si trascurano le dimensioni di un oggetto.

In definitiva, posto  $S = \{P, NP\}$ , una descrizione completa del comportamento dell'ascensore con gettoniera è data dalle due funzioni  $C : X \times S \rightarrow S$  e  $O : X \times S \rightarrow Y$  definite da

$$\begin{aligned} O(n,P) = Pn ; O(n,NP) = N ; O(A,P) = S ; O(A,NP) = S ; O(M,P) = N ; O(M,NP) = N \\ C(n,P) = NP ; C(n,NP) = NP ; C(A,P) = P ; C(A,NP) = NP ; C(M,P) = P ; C(M,NP) = P. \end{aligned}$$

Tali considerazioni conducono alla seguente definizione generale di macchina.

**Definizione 1.** Un *automa finito*<sup>2</sup> è costituito da:

- tre insiemi  $X, Y, S$  finiti, detti rispettivamente *insieme degli input*, *insieme degli output* ed *insieme degli stati*,
- uno stato  $s_0$  detto *stato iniziale*
- due funzioni  $C : X \times S \rightarrow S$  e  $O : X \times S \rightarrow Y$  dette rispettivamente *funzione di transizione* e *funzione di output*.

L'insieme  $S$  degli stati viene chiamato anche *memoria* dell'automa, e l'ipotesi di finitezza di  $S$  esprime il fatto che si intende studiare macchine a memoria finita. Le due funzioni  $C$  e  $O$  rappresentano il comportamento dell'automa nel senso che:

- se l'automa è nello stato  $s$  e riceve in entrata l'input  $x$
  - allora passa allo stato  $s' = C(x,s)$  e fornisce in uscita l'output  $y = O(x,s)$ .
- Più precisamente abbiamo la seguente definizione.

**Definizione 2.** Dato un automa, ad ogni sequenza  $x_1, \dots, x_n$  di inputs corrisponde una sequenza  $s_1, \dots, s_n$  di stati ed una sequenza  $y_1, \dots, y_n$  di outputs al modo seguente.

$$\begin{aligned} s_1 = C(x_1, s_0) & ; & y_1 = O(x_1, s_0) \\ s_2 = C(x_2, s_1) & ; & y_2 = O(x_2, s_1) \\ \dots & ; & \dots \\ s_n = C(x_n, s_{n-1}) & ; & y_n = O(x_n, s_{n-1}). \end{aligned}$$

**Esempio.** Riflettiamo sul modo usuale di calcolare la somma di due numeri, ad esempio i numeri 372 e 348. La prima cosa che viene fatta è incolonnare i due numeri, poi vengono lette le cifre da destra verso sinistra e, tenendo conto dell'eventuale riporto, vengono man mano scritte le cifre della somma

$$\begin{array}{r} 372+ \\ \underline{348} \\ 720 \end{array}$$

Un automa che volesse fare la stessa cosa, una volta che abbia ricevuto in input la sequenza (2,8), (7,4), (3,3) dovrebbe man mano stampare la sequenza 0,2,7. In tale automa sarebbe  $X =$

<sup>2</sup> Il termine *finito* è importante e corrisponde all'ipotesi per cui sono ammissibili solo un numero finito di input, di output e di stati. Nel capitolo prossimo parleremo anche della possibilità che gli insiemi  $X, Y$  e  $S$  siano numerabili. Sarebbe anche naturale considerare macchine in cui questi insiemi hanno la potenza del continuo. Ad esempio se consideriamo un'auto allora sono input accettabili il ruotare il volante in un certo modo, premere più o meno l'acceleratore, il frenare con maggiore o minore intensità. In risposta a tali input un'auto può dare come output una accelerazione. Tale output dipende dallo stato in cui è la macchina (velocità assunta in un dato istante, temperatura del motore, quantità di benzina nel serbatoio). In tutti i casi entrano in gioco grandezze che si possono rappresentare solo con numeri reali (posizione del volante, velocità, quantità di benzina ...).

$\{0, \dots, 9\} \times \{0, \dots, 9\}$  e  $Y = \{0, \dots, 9\}$ . E' chiaro inoltre che la risposta dell'automa deve dipendere non solo dalle cifre lette in input ma anche dell'esistenza di un eventuale riporto. Dobbiamo pertanto supporre che vi siano due possibili stati  $R$  e  $NR$  indicanti rispettivamente l'esistenza di un riporto o meno. Lo stato iniziale è  $NR$ . La funzione di output sarà definita al modo seguente:

$$O(0,0,NR) = 0, O(0,0,R) = 1, O(1,0,NR) = 1, O(1,0,R) = 2, O(1,1,NR) = 2, O(1,1,R) = 3, \\ \dots \quad O(8,8,NR) = 6, O(9,8,R) = 8, O(8,9,NR) = 7, O(9,9,R) = 9.$$

La funzione di cambiamento di stato sarà definita al modo seguente

$$C(0,0,NR) = NR, O(0,0,R) = NR, C(1,0,NR) = NR, C(1,0,R) = NR, C(1,1,NR) = NR, \dots \\ \dots \quad C(8,8,NR) = R, C(9,8,R) = R, C(8,9,NR) = R, O(9,9,R) = R.$$

Nel prossimo paragrafo, utilizzando la rappresentazione in base due, mostreremo come si possa costruire un automa addizionatore.

**Esempio.** Si consideri un automa in cui  $X = Y = \{0,1\}$ ,  $S = \{a,b\}$ ,

$$O(0,a) = 0, O(0,b) = 1, O(1,a) = 0, O(1,b) = 1.$$

$$C(0,a) = a, C(0,b) = a, C(1,a) = b, C(1,b) = a,$$

ed in cui lo stato iniziale sia  $b$ . In tale caso la sequenza 1 1 0 1 (letta da destra verso sinistra) si trasforma al modo seguente:

all'input 1 nello stato  $b$  corrisponde l'output 1 e nuovo stato  $a$  ;

" " 0 " "  $a$  " " 0 " "  $a$  ;

" " 1 " "  $a$  " " 0 " "  $b$  ;

" " 1 " "  $b$  " " 1 " "  $a$

Ottenendo la sequenza 1 0 0 1.

**Esercizio.** Con riferimento all'automa precedente, dire in che cosa si trasformano le sequenze 1 1 1, 0 0 0 e 0 1 0 .

La seguente ovvia proposizione mostra come la nozione di automa generalizza quella di funzione e che, in un certo senso, gli automi sono "funzioni con memoria" e le funzioni sono "automi senza memoria".

**Definizione 3.** Un automa ad un solo stato viene detto *automa senza memoria*. Un automa ad un solo possibile input viene chiamato *sistema isolato*.

Gli automi senza memoria coincidono con le funzioni di  $X$  in  $Y$ . Infatti se un automa ha un solo stato, diciamo  $s$ , allora la funzione  $C$  è necessariamente costante e non interviene nella determinazione della sequenza degli output. Allora l'automa computa la funzione  $f(x) = O(x,s)$ . Viceversa, data una funzione  $f$  di  $X$  in  $Y$ , consideriamo un automa in cui  $S$  contiene solo un elemento, diciamo  $s$  ed in cui  $O(x,s) = f(x)$ . E' immediato che tale automa computa la

funzione  $f$  e che alla sequenza di input  $x_1x_2\dots x_n$  fa corrispondere la sequenza di outputs  $f(x_1), \dots, f(x_n)$ .

Un sistema isolato è individuato da due funzioni di  $C : S \rightarrow S$  e  $O : S \rightarrow Y$ . Lo studio di un sistema isolato coincide con lo studio del comportamento di un qualunque automa a cui sia fornito input costante  $x$ . Possiamo interpretare il comportamento di un sistema isolato come “partenza”, in corrispondenza della prima volta in cui si fornisce l’input  $x$ , ed “evoluzione” in corrispondenza della successiva introduzione dell’input  $x$ . Riprenderemo l’esame dei sistemi isolati quando esamineremo le cose che un automa finito non può fare.

## 2. Macchine digitali (reti sequenziali)

Una classe molto importante di automi finiti è costituita dagli automi di tipo “digitale”. Infatti, come vedremo, ogni possibile automa finito può essere “simulato” da un automa digitale. L’idea fondamentale è quella di costruire automi in cui:

- gli input sono immessi tramite impulsi elettrici lungo  $n$  linee di entrata di un circuito in modo che, se l’input è rappresentato dalla  $n$ -pla  $x_1, x_2, \dots, x_n$  di elementi di  $\{0,1\}$ , allora:
  - se  $x_i = 1$  nella linea  $i$ -esima viene mandato un impulso,
  - se  $x_i = 0$  allora in tale linea non viene mandato alcun impulso.
- gli output sono forniti tramite impulsi lungo  $m$  linee di uscita assumendo come output la  $m$ -pla  $y = y_1, \dots, y_m$  di elementi di  $\{0,1\}$  con  $y_i = 1$  se nella linea  $i$ -esima viene emesso un impulso,  $y_i = 0$  altrimenti.
- gli stati sono registrati in  $h$  celle di memoria ciascuna delle quali può assumere solo due stati 0 ed 1 (corrispondenti, ad esempio, a cella non magnetizzata e cella magnetizzata).

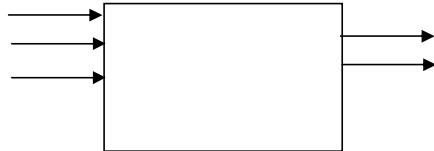
Chiameremo reti sequenziali un tale tipo di automi.

**Definizione 1.** Chiameremo *rete sequenziale* un automa tale che

$$X = \{0,1\}^n, Y = \{0,1\}^m \text{ e } S = \{0,1\}^h$$

con  $n, m$  ed  $h$  opportuni interi. Chiameremo *rete combinatoria* una rete sequenziale priva di memoria cioè una funzione di  $\{0,1\}^n$  in  $\{0,1\}^m$ .

Ad esempio nel caso  $n = 3$ ,  $m = 2$ , una rete sequenziale appare come un scatoletta con 3 linee di ingresso e due di uscita



Il numero  $h$  che caratterizza la memoria  $S$  è particolarmente importante in quanto è una misura della quantità di cose che una rete sequenziale può ricordare. La minima memoria (non nulla) si ottiene per  $h = 1$  ed in tale caso si dice che la memoria è di *un bit* (da *Binary digit*, ovvero cifra binaria). Un automa con un bit di memoria può assumere solo due stati interni che possiamo supporre immagazzinati in una cella di memoria sotto forma di 0 o di 1. Per eseguire l'addizione, come vedremo, è sufficiente un automa con un bit di memoria. Invece si dice che la memoria è di *un byte* se  $h = 8$ . Un byte è perciò composto da 8 celle di memoria, cioè da 8 bit consecutivi<sup>3</sup>.

0	1	1	1	0	0	1	0
---	---	---	---	---	---	---	---

8 celle di memoria per un byte

Una macchina che abbia la memoria di un byte può assumere  $2^8 = 256$  stati, cioè può ricordare 256 cose diverse. Un byte può essere visto anche come una parola di lunghezza 8 scritta nell'alfabeto  $\{0,1\}$ . Sono esempi di byte le parole 00101011, 11111000, .... Come vedremo, possiamo interpretare una tale parola anche come numero scritto in base 2. Nei computer attualmente in commercio ogni singolo carattere (lettera maiuscola, lettera minuscola, cifra, punteggiatura, spazio, ecc.) della tastiera viene rappresentato con un byte, in accordo con una tabella convenzionale detta ASCII (American Standard Code for Information Interchange). Prende il nome di *codice ASCII*, di un carattere il numero associato al byte che lo rappresenta. Ad esempio se in un calcolatore si tiene premuto il tasto *Alt* e si digita il numero 97 allora appare il carattere *a*. Se si digita il numero 98 allora si ottiene il carattere *b*.

Il byte viene considerato attualmente una unità di base per misurare la memoria di una disco o di un computer. Dal byte discendono poi i multipli KiB, MiB, GiB ...quali unità di misura della memoria (si usano per misurare sia la capacità di dischi e memorie, sia le dimensioni di file e cartelle). Ogni multiplo è  $1024 = 2^{10}$  volte il precedente<sup>4</sup>. Ad esempio un MiB

<sup>3</sup> A volte le parole *bit* e *byte* vengono usate non per denotare la memoria ma uno stato della memoria. Ad esempio si dice "un byte" per indicare una parola ottenuta come sequenza di otto cifre 0 ed 1. Come abbiamo già detto, si parla di "memoria di un byte" per indicare la possibilità di immagazzinare in memoria una qualunque di tali parole.

<sup>4</sup> E' molto frequente anche l'utilizzazione di multipli del byte secondo il rapporto  $10^3 = 1000$  del byte. In questo caso si utilizzano le sigle kB (kilobyte), MB (megabyte), GB (gigabyte) ... Poiché 1024 è quasi uguale a 1000, spesso si confondono le due misure.

corrisponde a 1.048.576 byte e quindi alla possibilità di scrivere, approssimativamente, un milione di parole dell'alfabeto usuale.

**Teorema 2.** Ogni automa finito può essere simulato, dopo una opportuna codifica binaria degli input, degli output e degli stati, da una rete sequenziale.

*Dim.* Sia  $\mathcal{A} = (\Sigma, Y, S, O, C, s_0)$  un qualunque automa finito. Essendo  $\Sigma$ ,  $Y$  ed  $S$  finiti, esisteranno degli interi  $n$ ,  $m$ ,  $h$  tali che la cardinalità di tali insiemi sia minore di  $2^n$ ,  $2^m$  e  $2^h$ , rispettivamente. Possiamo allora associare in maniera iniettiva

- ad ogni elemento  $\alpha$  di  $\Sigma$  una  $n$ -pla  $c_1(\alpha) \in \{0,1\}^n$
- ad ogni elemento  $y$  di  $Y$  una  $m$ -pla  $c_2(y) \in \{0,1\}^m$ ,
- ad elemento  $s$  di  $S$  una  $h$ -pla  $c_3(s) \in \{0,1\}^h$ .

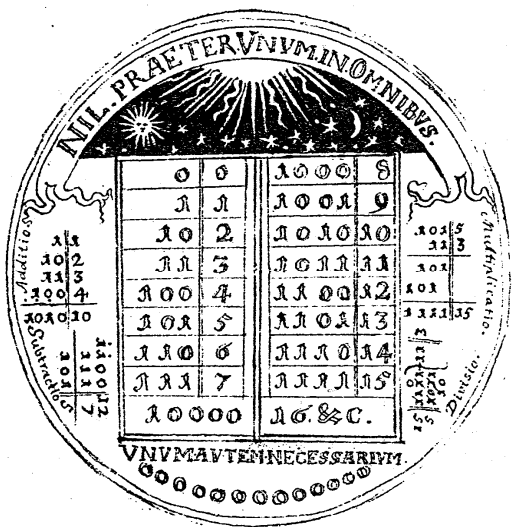
Tali codificazioni permettono di definire una rete sequenziale associando alla funzione  $O : \Sigma \times S \rightarrow Y$  una funzione  $O^* : \{0,1\}^n \times \{0,1\}^h \rightarrow \{0,1\}^m$  ottenuta ponendo  $O^*(a,b) = O(\alpha, s)$  se  $c_1(\alpha) = a$  e  $c_3(s) = b$  e definendo in un modo qualunque  $O^*(a,b)$  nel caso in cui  $a$  oppure  $b$  non risultano essere codici di qualche cosa. Allo stesso modo si definisce  $C^*$ . E' immediato che la rete sequenziale  $(\{0,1\}^n, \{0,1\}^m, \{0,1\}^h, O^*, C^*)$  è in grado di simulare l'automata  $\mathcal{A}$ .

Si noti che le codifiche  $c_1$ ,  $c_2$ ,  $c_3$  utilizzate nella dimostrazione del teorema non sono suriettive, in generale. Ad esempio, se il numero di elementi di  $\Sigma$  non è  $2^n$  ci saranno anche  $n$ -ple che non sono codici di elementi di  $\Sigma$ .

### 3. Leibniz: un mondo fatto di 0 ed 1.

*Esistono 10 tipi di persone: quelle che conoscono il sistema binario e quelle che non lo conoscono (Pseudo-Leibniz)*

Il primo che ha riconosciuto la possibilità di riferirsi solo alle cifre 0 ed 1 è stato il filosofo G. W. Leibniz. Leibniz, oltre ad essere, con Newton, il padre del moderno calcolo infinitesimale, è uno dei precursori della logica moderna. Fra le sue riflessioni nel campo del calcolo razionale occupa un posto particolare la scoperta del sistema binario di rappresentazione dei numeri, soprattutto per gli straordinari sviluppi che ha avuto nei



Di fianco riportiamo un disegno di Leibniz di un medaglione in cui nella tabella al centro sono scritti i numeri da 1 a 15 scritti in base due. Sulla sinistra è scritto un esempio di addizione ed uno di sottrazione. Sulla destra un esempio di moltiplicazione ed uno di divisione. Sono anche interessanti, da un punto di vista filosofico, le iscrizioni latine in cui la rappresentazione binaria viene messa in parallelo con la creazione in cui l' "uno" riesce a creare tutto dal "nulla", cioè dallo zero.

tempi attuali. Leibniz era molto interessato alla lingua cinese, di cui lo affascinava particolarmente la natura ideogrammatica. Un missionario della Compagnia di Gesù in Cina, appena tornato dall'Oriente, gli fece conoscere l'I Ching, il *Libro dei Mutamenti*, antichissimo testo usato come strumento di predizione del futuro. Il filosofo fu affascinato dal sistema di 64 esagrammi, ognuno dei quali è composto unicamente da due simboli: una linea spezzata e una intera. Leibniz attribuì a questo simbolismo un significato matematico che prima non aveva, infatti vi scorse un esempio di progressione di numeri binari, così come lui spiega nel suo saggio *Spiegazione della aritmetica binaria* (1703). Basta interpretare la linea spezzata come 0 e la linea intera come 1, allora gli esagrammi cinesi formano delle sequenze che possono essere lette come numeri scritti in forma binaria.

Il principale vantaggio della rappresentazione binaria è che l'informazione rappresentata in questo alfabeto è facilmente trattabile da opportuni dispositivi elettronici (si vedano i paragrafi successivi). Inoltre limitarsi a tale alfabeto non è affatto restrittivo. Ad esempio i numeri interi sono rappresentabili in modo adeguato utilizzando solo le cifre 0 ed 1. Per

mostrare come si possa fare questo, cominciamo con l'osservare che quando scriviamo il numero 325 utilizzando la "base 10" intendiamo il numero  $3 \cdot 100 + 2 \cdot 10 + 5 = 3 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0$ , cioè il numero composto da cinque unità, due decine e tre centinaia. Numeri più lunghi coinvolgeranno altre potenze di dieci. Ad esempio il numero  $23617 = 2 \cdot 10^4 + 3 \cdot 10^3 + 6 \cdot 10^2 + 1 \cdot 10^1 + 7 \cdot 10^0$  rappresenta 7 unità, più 1 di 10, più 6 di 100, più 3 di mille più 2 di diecimila. In definitiva ogni numero viene espresso come somma di potenze successive di 10. Siamo sicuri che tutti i numeri interi si possono rappresentare in questo modo? Ebbene l'usuale rappresentazione in base dieci dei numeri si basa sul seguente teorema di cui omettiamo la dimostrazione.

**Teorema 1.** Sia  $b$  un intero diverso da 1, allora per ogni numero naturale  $n$  esistono interi  $a_m, \dots, a_0$  minori di  $b$  tali che  $a_m \neq 0$  e  $n = a_m \cdot b^m + \dots + a_0 \cdot b^0$ . Inoltre tali interi sono unici.

E' possibile allora dare la seguente definizione.

**Definizione 2.** Sia  $b$  un numero intero diverso da 1 e sia  $A$  un insieme di  $b$  simboli in modo che ogni simbolo rappresenti uno dei numeri tra 0, 1, ...,  $b-1$ . Allora chiamiamo *rappresentazione in base  $b$*  del numero  $n$  la parola  $a_m \dots a_0$  in  $A$  con  $a_m \neq 0$ , tale  $n = a_m \cdot b^m + \dots + a_0 \cdot b^0$ .

Nel caso della usuale rappresentazione in base  $b=10$  allora si utilizzano le cifre 0, 1, ..., 9 per denotare i rispettivi numeri. Nel caso di rappresentazione in base 2 si utilizzano le cifre 0 ed 1. Ad esempio con la parola 10010 indicheremo il numero  $1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$ , cioè il numero composto da

- 0 unità,
- 1 gruppo di 2,
- 0 gruppi di 4,
- 0 gruppi di 8
- 1 gruppo di 16

che corrisponde al numero 18. Con la parola 10101 si denota il numero  $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$  che, scritto in base dieci, coincide con  $16 + 4 + 1 = 21$ . Naturalmente la parola 10101 e la parola 010101 rappresentano lo stesso numero. Pertanto se vogliamo che la rappresentazione sia univoca allora bisogna utilizzare solo la parola 0 e le parole che iniziano con 1.

Se si utilizza la base 3 allora si utilizzano solo le cifre 0, 1, 2. Ad esempio la stringa 1020 rappresenta il numero  $1 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3 + 0 \cdot 3^0$  cioè il numero che scritto in base dieci è uguale a 33. Se la parola 10101 che ho prima utilizzato per rappresentare in base 2 il numero 21 la interpretassi nella base 3 allora denoterebbe il numero  $1 \cdot 3^4 + 0 \cdot 3^3 + 1 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0$  il quale, scritto in base 10, coincide con  $81 + 9 + 1 = 91$ .

Possiamo anche considerare una base maggiore di dieci, ad esempio a volte si usa la base 16 che viene chiamata *esadecimale*. In questo caso oltre alle cifre 0, 1, ..., 9 si usano anche le lettere dell'alfabeto  $A, B, C, D, E, F$  per denotare rispettivamente i numeri 10, 11, 12, 13, 14 e 15. In tale caso, ad esempio, la parola  $A3B$  denota il numero  $A \cdot 16^2 + 3 \cdot 16^1 + B \cdot 16^0 = 10 \cdot 16^2 + 3 \cdot 16^1 + 11 \cdot 16^0 = 2560 + 48 + 11 = 2619$ .

**Metodo delle divisioni successive.** Fino ad ora abbiamo trasformato una rappresentazione in base diversa da dieci in quella in base dieci. E' possibile anche fare il viceversa con il *metodo delle divisioni successive*. Cominciamo con un esempio e rappresentiamo il numero decimale 57 in base 2 effettuando la divisione per 2 fino a che sia possibile

$$\begin{aligned} 57 : 2 &= 28 \text{ con resto } 1 \\ 28 : 2 &= 14 \text{ con resto } 0 \\ 14 : 2 &= 7 \text{ con resto } 0 \\ 7 : 2 &= 3 \text{ con resto } 1 \\ 3 : 2 &= 1 \text{ con resto } 1 \\ 1 : 2 &= 0 \text{ con resto } 1 \end{aligned}$$

Se si legge dal basso verso l'alto la colonna dei resti si ottiene la parola 111001 che è la rappresentazione in base 2 che cercavamo. D'altra parte le operazioni fatte mostrano proprio che:

$$57 = 2 \cdot 28 + 1, 28 = 2 \cdot 14, 14 = 2 \cdot 7, 7 = 2 \cdot 3 + 1, 3 = 2 \cdot 1 + 1$$

pertanto sostituendo e tentando di esprimere tutto come somma di potenze di due

$$57 = 2^2 + 2 + 1, 14 = 2^3 + 2^2 + 2, 28 = 2^4 + 2^3 + 2^2, 52 = 2^5 + 2^4 + 2^3 + 1$$

**Algoritmo dell'addizione.** Esaminiamo ora l'algoritmo dell'addizione in una base qualsiasi. base due. Nel caso della base 10, se ad esempio voglio addizionare 325 con 847 cioè  $3 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0$  con  $8 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0$  devo calcolare quante unità, decine centinaia e migliaia compaiono nel numero risultante. Mi accorgo allora che addizionando le unità ho una decina più due unità. Addizionando le decine ho sei decine più una di riporto, quindi sette decine, e così via. In definitiva si ha il solito algoritmo per l'addizione imparato dalle elementari:

$$\begin{array}{r} 325 \\ + 847 \\ \hline 1172 \end{array}$$

E' facile rendersi conto che l'algoritmo dell'addizione per numeri scritti in una base qualunque  $b$  è lo stesso di quello per la base 10. La differenza è che il riporto avviene quando la somma supera  $b$  e non quando supera 10. Ad esempio, con riferimento alla base 2, per eseguire l'addizione di utilizzano le seguenti tabelline:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$  con riporto di 1

Ad esempio:

$$\begin{array}{r} 11001 + \\ 1101 = \\ \hline 100110 \end{array}$$

Un altro esempio è il seguente:

$$10010 +$$

$$\begin{array}{r} 10011 \\ \underline{100101} \end{array}$$

Da notare che non è difficile costruire abachi che utilizzano la notazione in base  $b \neq 10$ . E' sufficiente porre in ogni linea dell'abaco  $b$  palline e poi applicare la stessa procedura utilizzata per la base 10.

**Moltiplicazione:** Si esegue la moltiplicazione tramite le seguenti tabelline:

- $0 \times 0 = 0$
- $1 \times 0 = 0$
- $0 \times 1 = 0$
- $1 \times 1 = 1$

Esempio

$$\begin{array}{r} 111001 \times \\ \quad 101 = \\ \hline 111001 \\ 000000 \\ 111001 \\ \hline 100011101 \end{array}$$

**Nota: la moltiplicazione degli Egiziani.** Il fatto che ogni numero può essere scritto in base due permette di effettuare la moltiplicazione tra due numeri con un metodo (inventato dagli antichi egiziani) che si basa sull'operazione di raddoppiare. Supponiamo che il problema sia di dovere moltiplicare 17 per 15, allora si costruiscono due colonne, la prima contenente le potenze successive di 2 minori di 17, la seconda numeri che si ottengono, a partire da 15, tramite successivi raddoppiamenti.

<u>1</u>	15
2	30
4	60
8	120
<u>16</u>	240

Poi si considerano i numeri della prima colonna che addizionati danno 17, cioè i numeri 1 e 16. Il prodotto cercato è dato dalla somma di 15 più 240 che corrispondono nella tabella ai numeri 1 e 16. Infatti

$$17 \times 15 = (1+16) \times 15 = (1+2^4) \times 15 = 15 + 2^4 \times 15.$$

Supponiamo ora di volere moltiplicare 11 per 13. Allora procediamo secondo la seguente tabella:

<u>1</u>	13
<u>2</u>	26

$$\begin{array}{r} 4 \quad 52 \\ 8 \quad 104 \end{array}$$

Essendo nella prima colonna 11 uguale la somma di 1, 2 e 8, il risultato sarà  $13+26+104=153$ . Infatti

$$11 \times 13 = (1+2+8) \times 13 = (1+2^1+2^3) \times 13 = 13+2 \times 13+2^3 \times 13.$$

Tale metodo si basa sul fatto che ogni numero può essere scritto come somme di potenze di due. Pertanto un prodotto del tipo  $x \times y$  se  $x = a_m \cdot 2^m + \dots + a_0 \cdot 2^0$  sarà sempre del tipo

$(a_m \cdot 2^m + \dots + a_0 \cdot 2^0) \times y = a_m \cdot 2^m y + \dots + a_0 \cdot 2^0 y$ . Osserviamo ora che i numeri  $a_m, \dots, a_0$  sono o uguali a zero oppure uguali ad 1 e che ovviamente dobbiamo riferirci nella somma solo a quelli uguali ad 1. Allora appare evidente che il prodotto che ci interessa è uguale alla somma di numeri che si ottengono raddoppiando più volte  $y$ .

#### 4. La logica degli Stoici per costruire un automa.

Il concetto di automa finito permette di studiare le macchine reali "a scatola nera" prendendo in esame solo il comportamento esterno della macchina. In questo paragrafo e nel successivo esamineremo come sia possibile progettare la struttura interna di un qualunque automa finito prefissato. Per tale scopo è essenziale fornire alcune nozioni di calcolo proposizionale. Una qualunque espressione di cui si possa dire che sia vera o falsa viene chiamata *proposizione*. Esempi di proposizione sono:

"ieri sono andato a Roma", " $2+3=7$ ", "7 è primo ed è maggiore di 5", "7 non è primo", "esiste un numero il cui quadrato è due", "34 o è pari oppure  $34-1$  è pari".

Non sono invece esempi di proposizione

"vai a Roma!", "vai a Roma?", "trova un numero il cui quadrato sia 2", " ~~$x+2=5$~~ ".

Nel seguito a volte invece di dire che una proposizione  $\alpha$  è vera diremo che ha "valore di verità 1", invece di dire che è falsa diremo che ha "valore di verità 0", scriveremo inoltre rispettivamente  $v(\alpha) = 1$  e  $v(\alpha) = 0$ . Ora vi sono alcune proposizioni che si ottengono a partire da altre proposizioni: chiameremo *composte* tali proposizioni mentre chiameremo *atomiche* le proposizioni che non sono composte. Il primo e secondo esempio di proposizione dato sopra costituiscono proposizioni atomiche. Il terzo esempio è una proposizione che può essere considerata composta (tramite congiunzione) dalla proposizione "7 è primo" e dalla proposizione "7 è maggiore di 5". Tale proposizione è vera perché tutte e due le componenti sono vere. Il quarto esempio è ottenuto (tramite negazione) dalla proposizione "7 è primo". Tale proposizione è falsa perché "7 è primo" è vera. Il quinto esempio di proposizione è ottenuto componendo (tramite disgiunzione) la proposizione "34 è pari" con la proposizione "34-1 è pari". Tale proposizione è vera perché una delle sue componenti è vera.

Compito del calcolo proposizionale è:

esaminare come il valore di verità delle proposizioni composte dipende dal valore di verità delle proposizioni componenti.

Procedendo in maniera più formale, consideriamo il linguaggio del calcolo proposizionale che si può definire come un linguaggio formale il cui alfabeto contiene:

i) un insieme  $V_p$  di  $n$  successione di elementi  $p_1, \dots, p_n, \dots$  chiamati variabili proposizionali che usualmente indichiamo con  $p_1, \dots, p_n, \dots$

ii) i simboli  $\wedge, \vee, \neg$  chiamati rispettivamente *congiunzione, disgiunzione, negazione ed implicazione*

iii) i due simboli  $( \text{ e } )$  chiamati *parentesi*.



$p_1$	$p_2$	$\neg p_2$	$p_1 \vee \neg p_2$	$\neg(p_1 \vee \neg p_2)$	$\neg(p_1 \vee \neg p_2) \rightarrow p_1$
1	1	0	1	0	1
1	0	1	1	0	1
0	1	0	0	1	0
0	0	1	1	0	1

Le tavole di verità delle formule  $p_1 \rightarrow p_2$  e  $p_1 \leftrightarrow p_2$  saranno

$p_1$	$p_2$	$p_1 \rightarrow p_2$
1	1	1
1	0	0
0	1	1
0	0	1

$p_1$	$p_2$	$p_1 \leftrightarrow p_2$
1	1	1
1	0	0
0	1	0
0	0	1

**Esercizio.** Scrivere le tavole di verità delle formule  $\neg(p_1 \rightarrow p_3)$  e  $(\neg p_3 \wedge p_2) \rightarrow p_2$ .

**Definizione 2.** Una *tautologia* (una *contraddizione*) è una proposizione che risulta vera (rispettivamente falsa) qualunque siano i valori di verità delle proposizioni componenti.

La formula  $p_1 \vee \neg p_1$  è una tautologia (nota sotto il nome di *terzo escluso*) mentre  $p_1 \wedge \neg p_1$  è una contraddizione (l'affermazione che una formula del tipo  $\alpha \wedge \neg \alpha$  è sempre falsa prende il nome di *principio di non contraddizione*).

**Esercizio.** Dire quali delle seguenti affermazioni è una tautologia:

- l'Italia è una monarchia
- o l'Italia è una monarchia oppure non lo è
- 5 è diverso da 0
- 5 è sia dispari che pari

**Definizione 3.** Due proposizioni sono *logicamente equivalenti* se assumono gli stessi valori di verità per ogni assegnazione di valori di verità delle proposizioni componenti (cioè se determinano la stessa tavola di verità).

Ad esempio le formule  $p_1 \vee p_2$  e  $p_2 \vee p_1$  pur essendo diverse (perché scritte diversamente), sono logicamente equivalenti. Ovviamente tutte le tautologie sono equivalenti tra loro e lo stesso vale per le contraddizioni.

**Esercizio.** Dire quali delle seguenti coppie di asserzioni sono logicamente equivalenti

- 5 è pari .... non è vero che 5 non sia pari
- sono promosso ..... ho avuto un voto maggiore o uguale a 6
- 5 è uguale a 0 .... l'Italia è una monarchia

**Nota.** Per come abbiamo data la definizione di tavola di verità associata ad una proposizione abbiamo che la formula  $(p_3) \vee (p_2)$  determina una funzione (costante rispetto alla prima variabile) definita in  $\{0,1\}^3$  anche se  $p_1$  non compare esplicitamente. Una situazione simile si manifesta quando si definisce il concetto di polinomio e di funzione polinomiale associata. Il polinomio  $2 \cdot (x_3)^4 + 5 \cdot (x_2)^2 + 7$  ad esempio determina una funzione polinomiale di  $R^3$  in  $R$  anche se la variabile  $x_1$  non compare esplicitamente. D'altra parte possiamo sempre supporre che sia presente anche il monomio  $0 \cdot x_1$ . Analogamente, nel caso della proposizione  $(p_3) \vee (p_2)$  possiamo sempre supporre che sia presente il "monomio"  $p_1 \wedge \neg p_1$  e quindi sostituire ad essa la proposizione logicamente equivalente  $((p_3) \vee (p_2)) \vee (p_1 \wedge \neg p_1)$ . Per lo stesso motivo alla proposizione  $(p_3) \vee (p_2)$  può essere associata anche una funzione di  $\{0,1\}^4$  in  $\{0,1\}$ . Basta identificare tale proposizione con la proposizione  $((p_3) \vee (p_2)) \vee (p_1 \wedge \neg p_1) \vee (p_4 \wedge \neg p_4)$ . Nel seguito quando dovremo descrivere una tavola di verità di una formula metteremo solo le colonne da cui dipendono effettivamente i valori della formula.

**Esercizio.** Dimostrare che le formule  $(\neg p_4) \wedge ((p_4) \vee (p_2))$  e  $(\neg p_4) \wedge (p_2)$  sono logicamente equivalenti.

**Esercizio.** Le formule  $(\neg p_4) \wedge (\neg p_5)$  e  $(\neg p_1) \wedge (\neg p_2)$  sono logicamente equivalenti?

Abbiamo visto che ogni proposizione determina una tavola di verità e quindi una funzione di  $\{0,1\}^n$  in  $\{0,1\}$ . Viceversa, ogni funzione di  $\{0,1\}^n$  in  $\{0,1\}$  può essere ottenuta come tavola di verità di una opportuna proposizione come mostra il seguente teorema noto sotto il nome di *teorema di completezza funzionale*.

**Teorema 4 (Teorema di completezza funzionale).** Per ogni funzione  $t : \{0,1\}^n \rightarrow \{0,1\}$  esiste una formula  $\alpha$  che ha  $t$  come tavola di verità. Pertanto i connettivi  $\wedge, \vee, \neg$  costituiscono un sistema completo.

*Dim.* Cominciamo con un esempio e consideriamo una tavola di verità  $t$  che assume solo il valore 0. Allora una qualunque contraddizione è una formula che ha  $t$  come tavola di verità. Ad esempio possiamo considerare la formula  $p_1 \wedge \neg p_1$  (interpretata come funzione di due variabili) o la formula  $(p_1 \wedge \neg p_1) \vee (p_2 \wedge \neg p_2)$ .

$p_1$	$p_2$	
1	1	0
1	0	0
0	1	0
0	0	0

$p_1$	$p_2$	
1	1	0
1	0	1
0	1	0
0	0	0

Consideriamo ora una tavola di verità che assume in una sola riga il valore 1, ad esempio nella seconda riga: allora è evidente che questa è la tavola di verità della  $p_1 \wedge \neg p_2$ . Tale formula si ottiene congiungendo il letterale  $p_1$  (in quanto il valore di  $p_1$  nella terza riga è 1) con il letterale  $\neg p_2$  (in quanto il valore di  $p_2$  nella terza riga è 0).

Consideriamo ora una tavola in cui viene assunto due volte il valore 1, ad esempio nella seconda e nella quarta riga: allora basta considerare la formula  $(p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge \neg p_2)$  che si ottiene come disgiunzione della formula  $p_1 \wedge \neg p_2$  (ottenuta guardando la seconda riga) con la formula  $\neg p_1 \wedge \neg p_2$  (ottenuta guardando la quarta riga).

$p_1$	$p_2$	
1	1	0
1	0	1
0	1	0
0	0	1

Più in generale:

- se nella tavola di verità compaiono  $h$  valori uguali ad 1 allora la formula cercata sarà la disgiunzione di  $h$  formule, una per ogni riga in cui compare 1.

Una dimostrazione più rigorosa è la seguente. Procediamo per induzione sul numero di volte  $u(t)$  in cui  $t$  assume il valore 1. Se  $u(t) = 0$  allora la contraddizione  $(p_1 \wedge (\neg p_1)) \vee \dots \vee (p_n \wedge (\neg p_n))$  ha  $t$  come tavola di verità. Se  $u(t) = 1$ , detto  $a = (a_1, \dots, a_n) \in \{0, 1\}^n$  l'unico elemento tale che  $t(a) = 1$ ,  $t$  sarà la tavola di verità della formula  $\alpha_1 \wedge \dots \wedge \alpha_n$  dove  $\alpha_i = p_i$  se  $a_i = 1$  e  $\alpha_i = \neg p_i$  se  $a_i = 0$ .

Sia  $u(t) > 1$  e sia  $a$  un elemento (cioè una riga) tale che  $t(a) = 1$ . Definiamo la tavola di verità  $t_1$  ponendo  $t_1(x) = t(x)$  se  $x \neq a$  e  $t_1(x) = 0$  se  $x = a$ . Indichiamo poi con  $t_2$  la tavola di verità che assume valore 1 solo in  $a$ . Poiché  $u(t_1) = u(t) - 1$ , per ipotesi di induzione esiste una formula  $\alpha_1$  la cui tavola di verità è  $t_1$ . Abbiamo già provato che, essendo  $u(t_2) = 1$  esiste una formula  $\alpha_2$  la cui tavola di verità è  $t_2$ . Osservando che  $t(x) = \max\{t_1(x), t_2(x)\}$ , possiamo concludere che  $t$  è la tavola di verità della formula in forma normale disgiuntiva  $\alpha_1 \vee \alpha_2$ .  $\square$

**Esempio.** Si considerino le seguenti tavole di verità

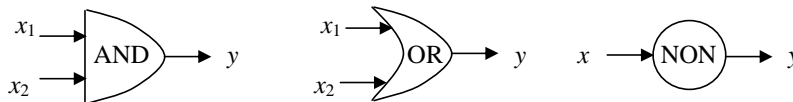
$p_1$	$p_2$	
1	1	1
1	0	0
0	1	1
0	0	0

$p_1$	$p_2$	
1	1	0
1	0	1
0	1	1
0	0	1

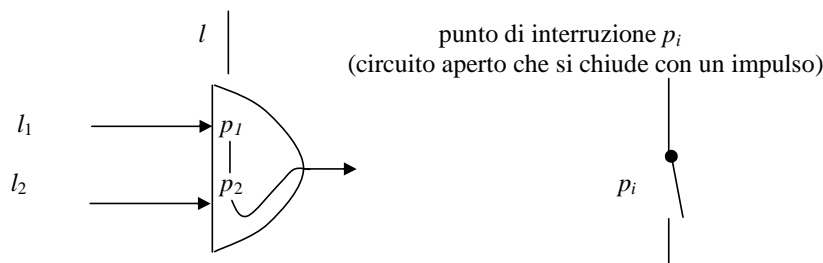
allora in base al teorema ora dimostrato sappiamo che tali tavole corrispondono rispettivamente alle formule  $(p_1 \wedge p_2) \vee (\neg p_1 \wedge p_2)$  e  $(p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2) \vee (\neg p_1 \wedge \neg p_2)$ .

### 5. Macchine digitali: porte logiche

In questo e nel prossimo paragrafo esaminiamo come potrebbero essere costruiti automi finiti di tipo "digitale". Associamo ad ogni connettivo logico una "porta logica" al modo seguente:

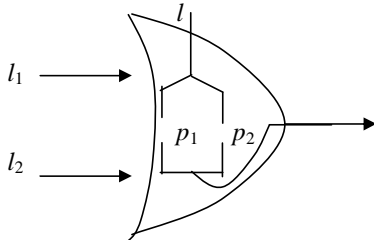


Il comportamento di una porta è determinato dalla tavola di verità del relativo connettivo. Ad esempio la porta AND emette un impulso (cioè  $y = 1$ ) solo se in entrambe le linee  $x_1$  ed  $x_2$  viaggia un impulso. La porta AND può essere realizzata da un circuito con due punti di interruzione in serie

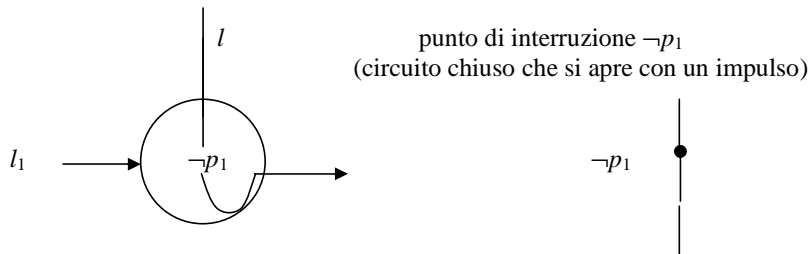


Nei punti in cui sono inserite le lettere  $p_1$  e  $p_2$  si devono immaginare degli interruttori. Tali interruttori sono comandati da impulsi che viaggiano lungo le linee  $l_1$  e  $l_2$ . Se in  $l_1$  viene mandato un impulso allora nel punto  $p_1$  il circuito *si chiude* e la corrente passa. Se in  $l_1$  non viene mandato un impulso allora nel punto  $p_1$  il circuito rimane aperto e la corrente non passa. La stessa cosa avviene per la linea  $l_2$  ed l'interruttore posto in  $p_2$ .

La porta OR viene invece realizzata tramite un circuito con due punti di interruzione *in parallelo*



La porta NOT viene realizzata al modo seguente

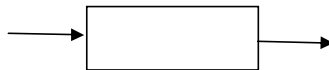


Nel punto di interruzione  $\neg p_1$  se non viene inviato nessun impulso la corrente passa, quando un impulso viene lanciato lungo la linea  $l_1$  si apre il circuito e la corrente non passa. Come vedremo

Fino ad ora abbiamo considerato le tre porte logiche relative ai tre connettivi  $\wedge$ ,  $\vee$ ,  $\neg$ . Come vedremo, tali porte siano sufficienti per costruire un qualunque automa finito. E' utile tuttavia definire altre porte. Ad esempio chiamiamo *porta ritardante* una porta che ha come effetto quello di non modificare il messaggio ricevuto ma di farlo ritardare di un tempo. Infatti si deve tenere conto dei tempi che impiegano gli impulsi in una rete nel "viaggiare" da sinistra verso destra. Una porta ritardante non cambia il segnale ed ha quindi come tavola di verità la seguente:

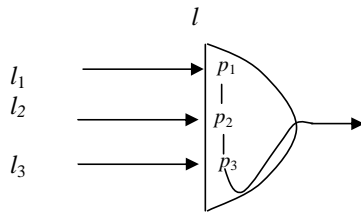
$p$	$p$
1	1
0	0

Indichiamo con un rettangolo tale porta logica,

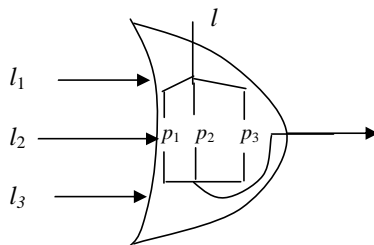


Potrebbe essere utile anche aggiungere nuove porte logiche. Ad esempio, per ogni intero  $n$ , una porta logica che realizza direttamente la *congiunzione di n formule*. In tale porta viene emesso un segnale solo se in tutte le  $n$  linee ingresso arriva un impulso. Possiamo realizzare una congiunzione a tre ingressi con tre punti di interruzione in serie





E' utile inoltre considerare, per ogni intero  $n$ , una porta logica relative alla *disgiunzione di  $n$  formule*. In tale porta viene emesso un segnale se in almeno una delle linee di entrata viene mandato un impulso. Nel caso di tre formule tale porta si realizza con tre punti di interruzione in parallelo:



L'utilizzo di tali porte rende molto semplice la rete che viene associata ad una tavola di verità.

## 6. Reti di porte logiche

Una *rete di porte logiche* è costituita da un insieme finito di porte logiche in cui le linee di uscita di alcune porte diventano linee di entrata in altre porte<sup>5</sup>. Le linee che non provengono da altre porte sono dette *linee input*. Le linee che non si immettono in altre porte sono dette *linee output*.

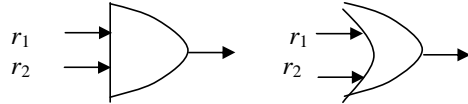
E' evidente che ogni rete di porte logiche con  $n$  linee di input ed  $m$  linee di output calcola una funzione di  $\{0,1\}^n$  in  $\{0,1\}^m$ . In particolare le reti con una sola linea di output calcolano tavole di verità.

**Teorema 1.** Ad ogni formula  $\alpha$  possiamo associare una rete di porte logiche  $r(\alpha)$  che computa la tavola di verità di  $\alpha$ . Ne segue che ogni tavola di verità può essere calcolata tramite una opportuna rete di porte logiche.

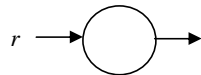
*Dim.* Definiamo, per ricorsione sulla complessità delle formule, una corrispondenza  $r$  che associa ad ogni formula  $\alpha$  una rete di porte logiche  $r(\alpha)$  che computa la tavola di verità di  $\alpha$ . A questo scopo date due reti con una sola linea di uscita  $r_1$  ed  $r_2$ , chiamiamo *congiunzione* e

<sup>5</sup> Fu Claude E. Shannon, uno dei fondatori della teoria dell'informazione, che nel 1937, nella sua tesi di Master in Electrical Engineering al MIT, stabilì per la prima volta l'analogia tra porte logiche, connessione in serie/parallelo di resistori ed i connettivi linguistici di disgiunzione e congiunzione.

disgiunzione di tali reti le reti, che indichiamo con  $r_1 \otimes r_2$  e con  $r_1 \oplus r_2$  definite in modo ovvio dalle seguenti immagini

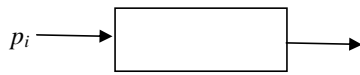


Chiamiamo *negazione* di una rete  $r$  la rete

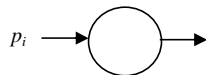


Denotiamo con  $n(r)$  una tale rete. Definiamo allora, data una formula  $\alpha$ , la rete  $r(\alpha)$  procedendo per induzione sulla complessità di  $\alpha$

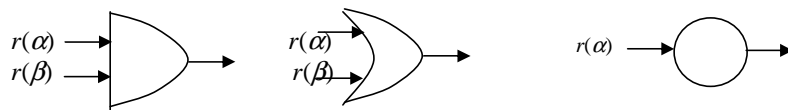
-  $r(p_i)$  è la rete



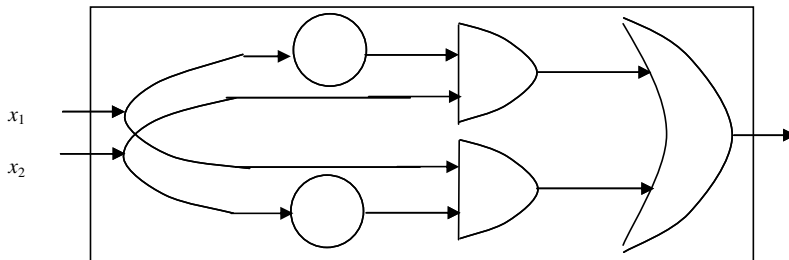
-  $r(\neg p_i)$  è la rete



-  $r(\alpha \wedge \beta) = r(\alpha) \otimes r(\beta)$ ,  $r(\alpha \vee \beta) = r(\alpha) \oplus r(\beta)$ ,  $r(\neg \alpha) = n(r(\alpha))$ , cioè  $r(\alpha \wedge \beta)$ ,  $r(\alpha \vee \beta)$  e  $r(\neg \alpha)$  sono le reti ottenute dalle reti  $r(\alpha)$  e  $r(\beta)$  al modo seguente:



Ad esempio, data la formula  $(\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2)$  otteniamo la rete



La corrispondenza  $r$  è definita per ogni possibile formula. Infatti se indichiamo con  $F$  l'insieme delle formule in cui  $r$  è definita, allora:

-  $F$  contiene le variabili proposizionali:

- se  $F$  contiene  $\alpha$  e  $\beta$  allora  $F$  contiene anche  $\alpha \wedge \beta$ ,  $\alpha \vee \beta$  e  $\neg \alpha$ .

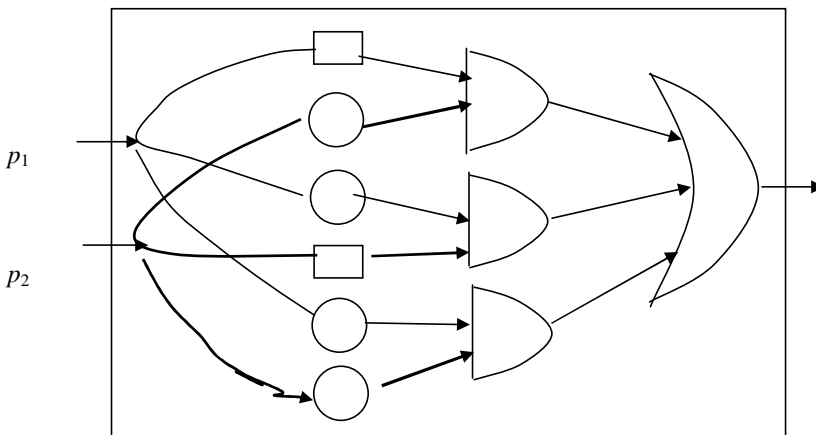
Da ciò segue che  $F$  contiene tutte le formule (si veda anche il paragrafo su ricorsione ed induzione nel capitolo 3).

Concludiamo osservando che è evidente che ogni formula  $\alpha$  ha una tavola di verità che è calcolabile tramite la rete  $r(\alpha)$ .

**Teorema 2.** Se si ammettono le porte ritardanti e quelle di disgiunzione e congiunzione generalizzate, allora ogni tavola di verità può essere realizzata da una rete a tre strati. L'ultimo strato è una porta di disgiunzione generalizzata, il penultimo è costituito da porte di congiunzione generalizzate, il primo strato da porte negazione o porte ritardanti.

*Dim.* Con il teorema di completezza abbiamo visto che ogni tavola di verità è la tavola di verità di una formula  $\alpha$  in forma normale disgiuntiva  $\alpha_1 \vee \dots \vee \alpha_n$  dove  $\alpha_i$  è la congiunzione di letterali. Se una rete per  $\alpha$  si ottiene costruendo reti per  $\alpha_1, \dots, \alpha_n$  e poi convogliando le uscite di tali reti in una porta logica di disgiunzione generalizzata. D'altra parte, poiché  $\alpha_i$  è una congiunzione di letterali, allora possiamo calcolare i valori di tale formula convogliando segnali di porte corrispondenti ai letterali in una congiunzione generalizzata. Infine i letterali si possono ottenere tramite porte ritardanti o negazioni. Ad esempio, supponiamo di volere trovare una rete di porte logiche che realizzi la tavola accanto. Allora una formula che abbia tale tavola come tavola di verità è  $(p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2) \vee (\neg p_1 \wedge \neg p_2)$ . La rete corrispondente, in cui si utilizza una porta per la disgiunzione a tre ingressi, è la seguente

$p_1$	$p_2$	
1	1	0
1	0	1
0	1	1
0	0	1



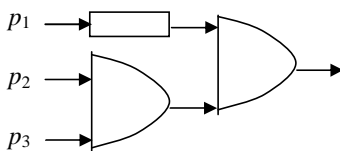
**Teorema 3.** Ogni rete combinatoria  $t : \{0,1\}^n \rightarrow \{0,1\}^m$  può essere realizzata da  $m$  reti di porte logiche ciascuna con  $n$  linee di ingresso.

*Dim.* La rete  $t : \{0,1\}^n \rightarrow \{0,1\}^m$  si identifica con le  $m$  funzioni di  $t_i : \{0,1\}^n \rightarrow \{0,1\}$  dove  $t_i(x_1, \dots, x_n)$  è la  $i$ -esima componente della  $m$ -pla  $t(x_1, \dots, x_n)$ . Pertanto il problema di realizzare  $t$  si traduce nel problema di realizzare  $m$  reti capaci di computare le tavole di verità  $t_i : \{0,1\}^n \rightarrow \{0,1\}$ .

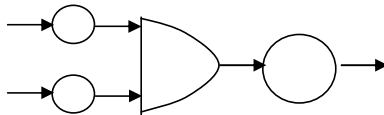
Riassumendo i risultati di questo paragrafo, abbiamo visto che se si vuole costruire un circuito digitale con  $n$  ingressi ed  $m$  uscite che esegua un certo compito (cioè che a certe entrate faccia corrispondere certe uscite) allora si può seguire la seguente procedura:

- si descrive il comportamento voluto tramite  $m$  tavole di verità  $t_1, \dots, t_m$  (ciascuna con  $n$  variabili),
- si costruiscono formule  $\alpha(1), \dots, \alpha(m)$  tali che  $t_{\alpha(i)} = t_i$ ,
- si costruiscono le reti  $r(\alpha(1)), \dots, r(\alpha(m))$
- si assemblano tali reti in modo da unificare gli ingressi.

**Nota: riduzione del numero di porte.** Le porte di congiunzione generalizzata possono essere realizzate tramite reti di porte congiunzione. Ad esempio quella a tre ingressi, che corrisponde a formule del tipo  $p_1(p_2 \wedge p_3)$ , può essere realizzata dalla rete



Un discorso simile può essere fatto per le porte di disgiunzione generalizzata. Sembrerebbero quindi sufficienti le sole porte relative ai tre connettivi logici. In effetti è possibile ridursi solo alle due porte logiche relative a AND e NOT. Infatti la formula  $\alpha \vee \beta$  è logicamente equivalente alla formula  $\neg(\neg\alpha \wedge \neg\beta)$  e quindi la porta logica di  $\vee$  può essere sostituita dalla rete



E' possibile anche utilizzare una unica porta logica per simulare qualunque rete combinatoria. Infatti introduciamo un nuovo connettivo logico  $\&$ , detto NAND, ed attribuiamo a tale connettivo la tavola di verità della formula  $\neg(p_1 \wedge p_2)$ .

$p_1$	$p_2$	$p_1 \& p_2$
1	1	0
1	0	1
0	1	1
0	0	1

Allora risulta che  $\neg p_1$  è equivalente a  $p_1 \& p_1$  e  $p_1 \wedge p_2$  è equivalente a  $(p_1 \& p_2) \& (p_1 \& p_2)$ . Se si definisce opportunamente una porta logica per tale tipo di connettivo allora ogni rete combinatoria può essere simulata da una rete di porte logiche tutte uguali a tale porta.

## 7. Macchine digitali: registri di memoria + reti di porte logiche

In questo paragrafo proviamo una fondamentale conseguenza del teorema di completezza funzionale:

tutti gli automi finiti possono essere realizzati utilizzando le semplici porte logiche AND, OR e NOT più opportuni registri di memoria.

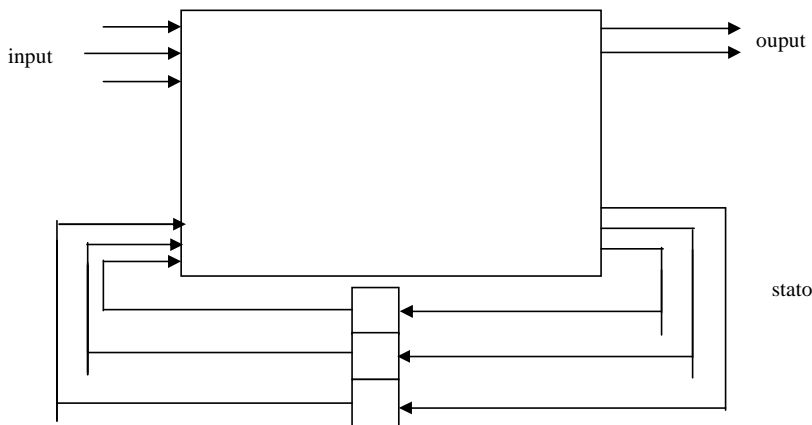
Il primo passo riguarda una classe particolare di automi finiti: le reti sequenziali.

**Teorema 1.** Ogni rete sequenziale può essere realizzata con una opportuna rete di porte logiche ed opportuni "registri di memoria".

*Dim.* Consideriamo una qualunque rete sequenziale e sia

$$\mathbb{X} = \{0,1\}^n, Y = \{0,1\}^m \text{ ed } S = \{0,1\}^h.$$

Allora le funzioni  $O : \mathbb{X} \times S \rightarrow Y$  e  $C : \mathbb{X} \times S \rightarrow S$  possono essere viste come una unica funzione di  $\mathbb{X} \times S$  in  $Y \times S$  o, se si vuole, come una rete combinatoria computante una funzione di  $\{0,1\}^{n+h}$  in  $\{0,1\}^{m+h}$ . Per quanto detto nella proposizione precedente, a tale rete corrisponderà un circuito con  $n+h$  linee di entrata ed  $m+h$  linee di uscita.



Se si aggiunge a tale circuito un registro di memoria con  $h$  celle che registrano gli output dalle linee  $s_1, \dots, s_h$  e forniscono input alle linee  $s_1, \dots, s_h$  si ottiene una "macchina" capace di realizzare la rete sequenziale voluta. Tale rete sequenziale funziona registrando per prima cosa lo stato iniziale (che è una  $h$ -pla di 0 ed 1) nel registro di memoria. Successivamente, quando viene fornito un input  $x$  (una  $n$ -pla di 0,1) nelle prime  $n$  linee di entrata, la rete combinatoria "legge" la rimanente parte di input  $s_1, \dots, s_h$  dal registro di memoria e lo immette nelle rimanenti  $h$  linee di entrata. Per quanto riguarda le uscite, le prime  $m$  linee forniscono l'output della rete sequenziale, le altre  $h$  linee inviano impulsi (o non-impulsi) al registro di memoria modificandone il contenuto e cambiando pertanto lo stato della rete sequenziale.

**Addizionale.** Esaminiamo ad esempio come si possa costruire un automa capace di effettuare la somma in base due in maniera sequenziale, cioè "leggendo" le cifre degli addendi da destra verso sinistra e fornendo, passo dopo passo, la corrispondente cifra del risultato

dell'addizione. In tale automa avremo che  $X = \{0,1\} \times \{0,1\}$ ,  $Y = \{0,1\}$ ,  $S = \{0,1\}$ , inoltre le funzioni di cambiamento di stato e di output sono funzioni definite in  $X \times S$  e quindi funzioni di tre variabili

$$O : \{0,1\} \times \{0,1\} \times \{0,1\} \rightarrow \{0,1\}, \quad C : \{0,1\} \times \{0,1\} \times \{0,1\} \rightarrow \{0,1\}$$

descritte dalla seguente tabella che descrive due tavole di verità

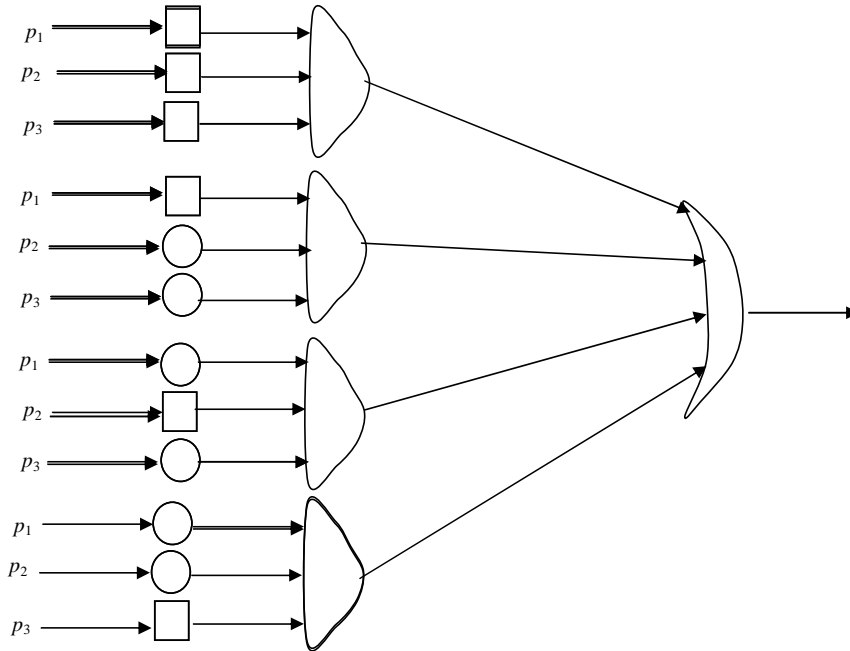
<i>prima cifra</i>	<i>seconda cifra</i>	<i>stato</i>	<i>Risultato</i>	<i>nuovo stato</i>
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

A queste due tavole di verità corrisponderanno le due formule

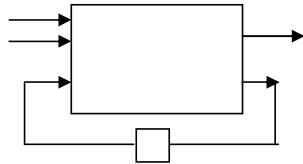
$$(p_1 \wedge p_2 \wedge p_3) \vee (p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge p_3);$$

$$(p_1 \wedge p_2 \wedge p_3) \vee (p_1 \wedge p_2 \wedge \neg p_3) \vee (p_1 \wedge \neg p_2 \wedge p_3) \vee (\neg p_1 \wedge p_2 \wedge p_3)$$

Alla prima formula corrisponderà la seguente rete a tre strati di porte logiche:



Similmente possiamo costruire una rete corrispondente alla seconda formula. Inserendo queste due reti all'interno del box, otterremo la seguente rete sequenziale.



**Teorema 2.** Ogni automa finito può essere realizzato tramite opportune reti di porte logiche e registri di memoria.

*Dim.* Nel Teorema 2 del paragrafo 2 abbiamo già mostrato come ogni automa finito possa essere simulato da una rete sequenziale. E' immediato che la rete sequenziale  $(\{0,1\}^n, \{0,1\}^m, \{0,1\}^h, O^*, C^*)$  è in grado di simulare l'automa  $\mathcal{A}$ . E' sufficiente allora applicare il Teorema 1.

Si noti che le codifiche  $c_1, c_2, c_3$  utilizzate nella dimostrazione del teorema non sono suriettive, in generale. Ad esempio, se il numero di elementi di  $\mathbb{X}^n$  non è  $2^n$  ci saranno anche  $n$ -ple che non sono codici di elementi di  $\mathbb{X}^n$ . Ciò non crea nessun problema come si vede nel seguente esempio.

**Esempio.** Supponiamo di avere un automa con  $\Sigma = \{a,b,c\}$ ,  $Y = \{m,n,o\}$ ,  $S = \{s,s'\}$ , e con le funzioni  $O$  e  $C$  descritte da

$$O(a,s) = m, O(a,s') = o, O(b,s) = n, O(b,s') = o, O(c,s) = m, O(c,s') = n;$$

$$C(a,s) = s, C(a,s') = s, C(b,s) = s', C(b,s') = s, C(c,s) = s', C(c,s') = s.$$

Per codificare  $\Sigma$  non è sufficiente  $\{0,1\}$  e pertanto dobbiamo ricorrere a  $\{0,1\}^2$ . Ad esempio, possiamo codificare  $a$  con 00,  $b$  con 01,  $c$  con 10. La coppia 11 non è codice di niente. Similmente, codifichiamo poi  $m$  con 00,  $n$  con 01 ed  $o$  con 10. Infine codifichiamo  $s$  con 0 ed  $s'$  con 1. Allora il nostro automa può essere realizzato tramite la rete sequenziale definita dal porre  $\Sigma = Y = \{0,1\}^2$ ,  $S = \{0,1\}$  e

$$O(00,0) = 00, O(00,1) = 10, O(01,0) = 01, O(01,1) = 10, O(10,0) = 00, O(10,1) = 01;$$

$$C(00,0) = 0, C(00,1) = 0, C(01,0) = 1, C(01,1) = 0, C(10,0) = 1, C(10,1) = 0.$$

Per completare la descrizione della rete si deve naturalmente dire anche quale è il suo comportamento con l'input 11. Ciò anche se tale input non si presenta mai perché nell'automata di partenza non esiste un input che si codifica con tale coppia. Ciò significa anche che non importa in quale modo si debba definire tale comportamento. Ad esempio possiamo porre  $O(11,0) = 00$ ,  $O(11,1) = 11$ ,  $C(11,0) = 0$ ,  $C(11,1) = 1$ . Abbiamo pertanto due tavole di verità per la funzione  $O$  ed una tavola di verità per  $C$ .

$p_1$	$p_2$	$p_3$	$O$	$O$	$C$
1	1	1	0	0	0
1	1	0	0	0	0
1	0	1	0	1	0
1	0	0	0	0	1
0	1	1	1	0	0
0	1	0	0	1	1
0	0	1	1	0	0
0	0	0	0	0	0

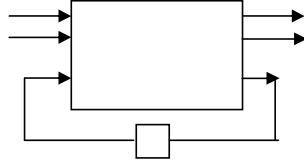
A queste tre tavole di verità corrisponderanno le tre formule

$$(\neg p_1 \wedge p_2 \wedge p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge p_3);$$

$$(p_1 \wedge \neg p_2 \wedge p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3);$$

$$(p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3).$$

A queste tre formule corrisponderanno tre circuiti che, assemblati, forniranno la seguente rete sequenziale:



### 8. Cose che un automa non potrà mai fare

Un modo di rapportare gli automi alle funzioni è quello di coinvolgere al posto di  $X$  ed di  $Y$  gli insiemi di parole  $X^*$  e  $Y^*$ . Infatti, un automa determina una funzione di  $X^*$  in  $Y^*$ ; in tale caso diremo che tale funzione è computabile tramite tale automa. In termini di teoria dei linguaggi formali, possiamo dire che un automa muta ogni parola nell'alfabeto  $X$  in una parola nell'alfabeto  $Y$ . Naturalmente parole corrispondenti hanno la stessa lunghezza ma una tale limitazione pur essere eliminata utilizzando input ed output nulli. Come vedremo, non tutte le funzioni di  $X^*$  in  $Y^*$  sono computabili tramite un automa finito. Premettiamo la seguente proposizione.

**Teorema 1.** Supponiamo che gli input forniti ad un automa ad  $n$  stati siano sempre uguali. Allora dopo un certo numero finito  $h < n$  di passi la sequenza degli stati e quella degli output entra in un ciclo di lunghezza finita  $p$ . Risulta inoltre che  $h+p \leq n$ .

*Dim.* Partiamo dallo stato iniziale  $s_0$ , forniamo input costante  $x$  ed indichiamo con

$$s_1 = C(x, s_0), \dots, s_j = C(x, s_{j-1}) \dots$$

la successione degli stati che l'automata man mano assume. Poiché l'insieme degli stati possibili dall'automata è finito, tale successione non può avere indefinitamente elementi diversi tra loro (al più ve ne possono essere  $n$  diversi tra loro). Pertanto ad un certo istante verrà assunto uno stato  $s_j$  che coincide con uno stato  $s_h$  già assunto precedentemente,  $s_h = s_j$ , con  $h \leq j \leq n$ . Allora, posto  $p = j-h$ , risulterà

$$s_h = s_j = s_{h+p}$$

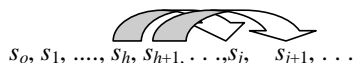
$$s_{h+1} = C(x, s_h) = C(x, s_j) = s_{j+1}$$

$$s_{h+2} = C(x, s_{h+1}) = C(x, s_{j+1}) = s_{j+2}$$

...

$$s_{h+p} = C(x, s_{h+p-1}) = C(x, s_{j+p-1}) = s_{j+p} = s_{h+2p}$$

Pertanto dall'equazione  $s_h = s_j$  segue che la sequenza  $s_h s_{h+1} \dots s_j$  è uguale alla sequenza  $s_j s_{j+1} \dots s_{j+p}$ . Per lo stesso motivo dall'equazione  $s_{h+p} = s_{h+2p}$  segue che la sequenza  $s_{h+p} \dots s_{h+2p}$  è uguale alla sequenza  $s_{h+2p} \dots s_{h+3p}$ . In definitiva la macchina dopo avere assunto nella sua evoluzione una sequenza finita di stati  $s_0, s_1, \dots, s_h$ , dal punto  $s_h$  in poi entra in un ciclo  $s_h, s_{h+1}, \dots, s_j$  di lunghezza  $p$  che si ripete indefinitamente.



Per quanto riguarda gli output, è sufficiente osservare che, stante il fatto che gli input coincidono sempre con  $x$ , per ogni indice  $i$  risulta  $y_i = O(x, s_i)$ . Allora poiché gli stati entrano in un ciclo finito, anche gli output si ripetono ciclicamente.  $\square$

Nella proposizione ora provata si può sostituire l'ipotesi di input costante con quella di nessun input. Possiamo pertanto provare che un automa "isolato" dal mondo esterno prima o poi finisce col cadere in un ciclo finito.

**Teorema 2.** Supponiamo di avere una macchina (un organismo, un sistema) tale che:

- i) può assumere solo un numero finito di stati
- ii) il comportamento in un istante  $h+1$  dipenda solo dallo stato della macchina all'istante precedente  $h$  (comportamento determinista)
- iii) la macchina sia isolata, cioè non riceva input dal resto dell'universo.

Allora la storia della macchina prima o poi entrerà in un ciclo che si ripete all'infinito<sup>6</sup>.

*Dim.* Si può ripetere il ragionamento della proposizione precedente. Consideriamo l'evoluzione degli stati da un istante in poi,  $s_0, s_1, \dots$ . Allora l'ipotesi di "comportamento determinista" comporta che lo stato  $s_{i+1}$  è determinato completamente dallo stato precedente secondo una legge data, e quindi che esiste una funzione  $f$  tale che  $s_{i+1} = f(s_i)$ . L'ipotesi di finitezza comporta che la macchina, dopo avere assunto un certo numero di stati, debba assumere necessariamente uno stato già assunto in precedenza. Quindi esiste  $s_j$  tale che  $s_j = s_h$  con  $h \leq j$ . Ma allora, posto  $p = j-h$ ,

$$\begin{aligned} s_{h+1} &= f(s_h) = f(s_j) = s_{p+h+1} \\ s_{h+2} &= f(s_{h+1}) = f(s_{p+h+1}) = s_{p+h+2} \\ &\dots \\ s_{h+p} &= f(s_{h+p-1}) = f(s_{p+h+p-1}) = s_{2p+h} \\ &\dots \end{aligned}$$

$\square$

**Teorema 3.** Se  $r$  è un numero irrazionale, non esiste nessun automa capace di stampare indefinitamente tutte le cifre di  $r$ .

<sup>6</sup> Naturalmente tale teorema potrebbe essere applicato anche all'universo in cui viviamo (se se ne accetta la finitezza ed il determinismo) e comporterebbe che la storia si ripete ciclicamente. In questo caso se si accetta il determinismo ogni stato  $s_{i+1}$  dell'universo dipende in maniera determinista dallo stato precedente  $s_i$ . Se si accetta che l'universo può assumere solo un numero di stati finiti allora ... Il tema dell' "eterno ritorno" ha occupato l'interesse di molti filosofi. Ad esempio Nietzsche dice: «Che accadrebbe se, un giorno o una notte, un demone strisciasse furtivo nella più solitaria delle tue solitudini e ti dicesse: "Questa vita, come tu ora la vivi e l'hai vissuta, dovrai viverla ancora una volta e ancora innumerevoli volte, e non ci sarà in essa mai niente di nuovo, ma ogni dolore e ogni piacere e ogni pensiero e sospiro, e ogni indicibilmente piccola e grande cosa della tua vita dovrà fare ritorno a te, e tutte nella stessa sequenza e successione [...]. L'eterna clessidra dell'esistenza viene sempre di nuovo capovolta e tu con essa, granello della polvere!» Nietzsche, *La gaia scienza*.

*Dim.* Infatti supponiamo di impartire ad una macchina l'ordine di stampare una dopo l'altra le cifre di  $r$ . Dopo che l'ordine è stato impartito la macchina non riceve altri ordini e quindi si trova in stato di isolamento. Ne segue che dopo avere stampato una sequenza finita di cifre di  $r$  la macchina deve, per il teorema 2.2, cadere in un ciclo infinito. Ciò è in contrasto con il fatto che, essendo  $r$  irrazionale, la successione delle sue cifre non presenta un andamento periodico.  $\square$

Ad esempio, non esiste un automa capace di calcolare le radici quadrate.

**Nota:** Che cosa succede circa la possibilità di stampare le cifre di un numero razionale? Esaminiamo il noto algoritmo della divisione. Ad esempio effettuiamo il calcolo di  $11/7$

$$\begin{array}{r}
 11 \quad | \overline{7} \\
 4 \quad 1.5714285 \dots \\
 \quad 5 \\
 \quad \quad 1 \\
 \quad \quad \quad 3 \\
 \quad \quad \quad \quad 2 \\
 \quad \quad \quad \quad \quad 6 \\
 \quad \quad \quad \quad \quad \quad 4 \\
 \quad \quad \quad \quad \quad \quad \quad 5 \\
 \quad \quad \quad \quad \quad \quad \quad \quad \dots
 \end{array}$$

Se si analizza tale algoritmo ci si accorge che la quantità di memoria utilizzata è finita. Indichiamo con  $r_0 = 4, r_1 = 5, \dots$ , la successione dei resti e con  $y_1 = 5, y_2 = 7, \dots$  la successione delle cifre. Ad ogni passo la cifra stampata  $y_{i+1}$  (output) dipende solo dal resto  $r_i$  precedentemente calcolato (stato). Più precisamente, fissato il numero 7 possiamo definire un automa con sette stati  $0, 1, 2, 3, 4, 5, 6$  e dieci possibili output (le cifre). Dato lo stato  $r_i$  al passo successivo si divide  $r_i$  per 7, si stampa la parte intera di tale divisione e si assume come stato il nuovo resto di tale divisione. Non vale la pena descrivere in modo dettagliato tale automa, tuttavia guardare la divisione da questo punto di vista permette di capire perché le cifre dell'espansione decimale di un numero razionale si sviluppano secondo un periodo ed un antiperiodo. Vedendolo direttamente, poiché i resti possibili sono  $0, 1, \dots, 6$ , dopo al più 6 passi si deve ottenere un resto che si era ottenuto già prima. Nel nostro caso il resto 4. Da questo momento in poi è evidente che si entra in un ciclo e che i resti e le cifre dell'espansione decimale si ripetono ciclicamente.

In uno degli esempi precedenti abbiamo visto che esiste un automa finito capace di effettuare l'addizione in maniera sequenziale. La proposizione seguente mostra che invece non esiste nessun automa finito capace di effettuare la moltiplicazione.

**Teorema 4.** Non esiste un automa finito capace di eseguire la moltiplicazione in maniera sequenziale (senza limiti al numero di cifre degli input).

*Dim.* Supponiamo per assurdo che esista un automa capace di effettuare qualsiasi moltiplicazione tra numeri naturali in maniera sequenziale, cioè "leggendo" le cifre dei

numeri da moltiplicare da destra verso sinistra e fornendo la cifra corrispondente del prodotto di tali numeri. Tale automa avrà come insieme di input  $X = \{0, \dots, 9\} \times \{0, \dots, 9\}$  e come insieme di output  $Y = \{0, \dots, 9\}$ . Poiché il numero di cifre del prodotto è superiore al numero di cifre dei due numeri da moltiplicare, sarà necessario anche un tasto-input, chiamiamolo  $a$ , che dica all'automata di andare avanti dopo che gli sono stati forniti tali numeri. Supponiamo che tale automa abbia  $n$  stati e proponiamoci di fargli eseguire moltiplicazioni del tipo  $10^m$  per  $10^m$ . Per fare questo dobbiamo fornire come input le coppie  $(0,0)$   $m$  volte, poi la coppia  $(1,1)$  e poi la lettera  $a$  un numero opportuno di volte. Se  $s(m)$  è lo stato assunto dall'automata dopo le  $m$  coppie  $(0,0)$ , è chiaro che se  $m \neq m'$  allora  $s(m) \neq s(m')$ . Infatti altrimenti il risultato della moltiplicazione di  $10^m$  per  $10^m$  coinciderebbe con il risultato della moltiplicazione di  $10^m$  per  $10^{m'}$ . Ne segue che la funzione  $s$  è iniettiva e quindi che la cardinalità dell'insieme degli stati è infinita, in contrasto con l'ipotesi che si tratti di un automa a stati finiti. In altre parole,

- per potere moltiplicare  $10$  per  $10$  l'automata ha bisogno di 1 stato almeno
- per potere moltiplicare  $10^2$  per  $10^2$  l'automata ha bisogno di almeno 2 stati diversi
- ...
- per potere moltiplicare  $10^m$  per  $10^m$  l'automata ha bisogno di almeno  $m$  stati diversi tra loro
- ...

Pertanto l'automata per potere effettuare tutte le moltiplicazioni dovrebbe avere infiniti stati.  $\square$

**Teoremi limitativi.** Le proposizioni che abbiamo provato sono esempi di "teoremi limitativi" in informatica. Tali teoremi individuano alcune cose che una macchina a memoria finita non può fare e richiedono qualche chiarimento. Per prima cosa, riferendoci ad esempio al problema della moltiplicazione, ciò che abbiamo dimostrato è che non esiste un unico automa capace di effettuare la moltiplicazione qualunque sia il numero di cifre dei numeri da moltiplicare. Non è difficile invece vedere che per ogni intero  $n$  esiste un automa capace di moltiplicare numeri con meno di  $n$  cifre; d'altra parte tali automi sono quelli comunemente in commercio.

Sofferamoci poi sulla espressione "non esiste" che appare nell'enunciato dei teoremi limitativi. Essa non deve essere intesa semplicemente nel senso che "al giorno d'oggi nessun è stato capace di costruire" un automa del tipo voluto. Piuttosto essa deve intendersi nel senso che "è impossibile che un tale automa si possa mai costruire", per quanti progressi possa fare la tecnologia.

Si noti che il primo e più famoso teorema limitativo in matematica è quello che asserisce che non esiste nessun numero razionale esprime la misura della diagonale di un quadrato di lato unitario. In termini algebrici, tale teorema asserisce che non esiste un numero razionale uguale alla radice di 2 (ovviamente, non dice che fino ad ora nessuno è stato capace di trovare un numero razionale uguale alla radice di 2). Altri teoremi limitativi sono quelli relativi alle costruzioni con riga e compasso viste nel capitolo precedente.

### 9. Considerazioni sugli automi finiti.

Abbiamo visto come non esistano automi capaci di fare l'estrazione della radice quadrata di 2 oppure il prodotto di due numeri indipendentemente dalla grandezza dei numeri che entrano in gioco. Ciò appare alquanto strano se si pensa alla semplicità di tali operazioni. Il problema risiede nella ipotesi di finitezza della memoria che abbiamo fatto per gli automi. Per renderci

conto di tale problema, cerchiamo di capire perché per la moltiplicazione si presentano problemi che invece non si presentano per l'addizione. A tale scopo basta confrontare il procedimento "carta e penna" che si insegna ai bambini per fare l'addizione di due numeri a più cifre con l'analogo procedimento per fare la moltiplicazione. Nel primo caso siamo in grado di effettuare l'operazione mentalmente, leggendo man mano gli addendi e dicendo il risultato. Infatti possiamo servirci della nostra memoria per ricordare l'eventuale riporto che di volta in volta si presenta. Lo sforzo di memoria per effettuare tale operazione non dipende dalla lunghezza dei numeri che entrano in gioco.

Nel secondo caso invece se i numeri hanno più di due cifre non riusciamo più a ricordare i calcoli intermedi ed abbiamo bisogno di scrivere su di un foglio i risultati di tali calcoli. Certamente una persona dotata di molta memoria pur tentare di effettuare i calcoli mentalmente per numeri di due cifre, di tre cifre o più. Purtroppo però è necessaria tanta più memoria quanto più grandi sono i numeri da moltiplicare. Infatti un esame del ben noto algoritmo di moltiplicazione mostra che dobbiamo scrivere (equivalentemente ricordare) tante righe quante sono le cifre del secondo numero da moltiplicare e quindi lo spazio richiesto (la memoria richiesta) è tanto più grande quanto più grandi sono i due numeri da moltiplicare. In definitiva, se (come avviene per gli automi finiti) si fissa a priori la quantità di memoria a disposizione (numero di righe da occupare e lunghezza delle righe) allora è possibile moltiplicare solo numeri con un numero di cifre prefissato.

Lo stesso discorso vale se si analizza, ad esempio, il procedimento di estrazione della radice di due. Quante più cifre di tale numero irrazionale si vogliono quanta più memoria (numero di righe) risulta necessaria.

**La definizione proposta di automa è ragionevole ?** In psicologia il *comportamentismo* è il punto di vista in cui non si cerca di descrivere i meccanismi psicologici che conducono un animale o persona ad un certo comportamento. Ci si limita a descrivere tale comportamento in corrispondenza di diversi stimoli o situazioni da parte dell'ambiente. Si esamina un animale oppure una persona come fosse un topolino da laboratorio in modo non differente da come si potrebbe esaminare una reazione chimica o i risultati di un esperimento di fisica. In definitiva ci si riferisce ad un paradigma di tipo input-output. La nozione di automa che abbiamo proposto si rifà a questo punto di vista. Supponiamo di avere una macchina davanti a noi e di volerla in qualche modo studiare descrivendo solo il suo comportamento, senza entrare nel merito dei meccanismi interni che la compongono. Allora si individua l'insieme  $X$  degli input che la macchina può ricevere e l'insieme delle risposte  $Y$  (output) che la macchina può dare. Se la macchina è molto semplice, allora basta costruire una tabella input-output per la completa descrizione della macchina e quindi una funzione  $f: X \rightarrow Y$ . Tuttavia in generale il comportamento della macchina non dipende solo dall'input fornito ad un dato istante ma anche dallo stato della macchina. Supponendo che gli stati della macchina si possano leggere in qualche modo, possiamo descrivere completamente il comportamento della macchina tramite una tabella che rappresenta la funzione  $O: X \times S \rightarrow Y$  ed un'altra tabella che rappresenta la funzione  $C: X \times S \rightarrow S$ .

**E se l'uomo fosse un automa ?** Che gli animali fossero automi era un convincimento, ad esempio, di Cartesio (si veda lo scritto di Cartesio alla fine del capitolo). Cartesio tuttavia

collocava l'uomo in una posizione diversa distinguendolo dagli animali. Un matematico che ha posto la questione se una macchina possa avere lo stesso comportamento intelligente di un uomo è proprio il fondatore della moderna informatica teorica, A. Turing. E' famoso il *Test di Turing* cioè un test che proponeva Turing come verifica dell'ipotesi che sia possibile costruire una macchina con la stessa intelligenza di un uomo. Sostanzialmente tale testa afferma che: è possibile riprodurre l'intelligenza umana se e solo se è possibile costruire un programma in grado di rispondere a domande che gli vengono fatte (supponiamo per internet e per iscritto) Se non siamo in grado di accorgerci se dall'altra parte ci sia un uomo oppure no, allora si è avuta una riproduzione perfetta di un uomo.

Naturalmente la questione se l'uomo o gli animali siano delle macchine (magari a base chimica) è enormemente difficile da affrontare ed è oltre gli scopi di questi appunti. Tuttavia è interessante (e divertente) fare un po' di fantamatematica ed esaminare quali conseguenze avrebbe l'ipotesi

Uomo = macchina molto complicata

Eccone alcune:

*Un uomo può essere trasmesso per internet.*

Infatti un automa è un oggetto finito, in un certo senso un programma al computer. Pertanto è possibile trasmettere, dopo una opportuna codifica, un automa via internet. Per lo stesso motivo varrebbero anche le seguenti affermazioni

*Un uomo può essere registrato su un dischetto*

*Un uomo può essere trasmesso per via radio*

*Un uomo può viaggiare nello spazio alla velocità della luce.*

*Un uomo può vivere in eterno in quanto può essere ricreato alla fine della sua vita in modo uguale*

*Di un uomo possono essere fatte quante copie si vuole*

*Se un uomo si ammala è molto più economico comprarne un altro (visto le parcelle dei medici)*

*Se ti senti poco bene è conveniente fare una copia di sicurezza di te stesso ...*

### 10. Il gruppo delle porte reversibili.

E' possibile costruire porte logiche che implementano direttamente reti combinatorie, cioè funzioni  $t : \{0,1\}^n \rightarrow \{0,1\}^m$ . In particolare, siamo interessati al caso in cui  $t$  sia invertibile (e questo ha senso solo se  $n = m$ ).

**Definizione 1.** Una *rete combinatoria reversibile* ad  $n$  ingressi è una funzione invertibile  $t : \{0,1\}^n \rightarrow \{0,1\}^n$ .

Nel caso  $n = m = 1$  le uniche funzioni invertibili di  $\{0,1\}$  in  $\{0,1\}$  sono l'applicazione identica (che corrisponde alla formula atomica  $p_1$ ) e la funzione  $1-x$  che abbiamo utilizzato per interpretare la negazione (che corrisponde alla formula  $\neg p_1$ ). Per  $n$  maggiore di 1 non esiste una interpretazione logica delle tavole delle reti combinatorie invertibili. Il seguente è un esempio di una rete invertibile  $t : \{0,1\}^2 \rightarrow \{0,1\}^2$  in cui abbiamo indicato con  $(p_1, p_2)$  l'input e con  $(q_1, q_2)$  l'output:

$p_1$	$p_2$	$q_1$	$q_2$
1	1	1	0
1	0	0	0
0	1	1	1
0	0	0	1

Poiché  $t$  è iniettiva non compaiono nelle ultime due colonne due coppie uguali.

Una *porta logica reversibile* è una realizzazione fisica di una rete combinatoria reversibile ed è quindi un apparecchio con  $n$

entrate ed  $n$  uscite. Ricordiamo ora una classe di gruppi particolarmente importante.

**Proposizione 2.** Sia  $S$  un insieme, allora la classe delle funzioni invertibili di  $S$  in se stesso è un gruppo  $G(S)$  rispetto alla operazione di composizione. L'elemento neutro di tale gruppo è l'applicazione identica, l'inverso di un elemento  $f \in G(S)$  è la funzione  $f^{-1}$  inversa di  $f$ .

Tale gruppo prende il nome di *gruppo delle permutazioni* su  $S$ . In particolare abbiamo che l'insieme delle reti reversibili ad  $n$  ingressi costituisce un gruppo.

**Definizione 3.** Chiamiamo *n-gruppo delle reti reversibili* il gruppo  $G_n$  delle permutazioni sull'insieme  $\{0,1\}^n$ .

L'importanza delle porte reversibili è mostrata dal seguente teorema. Ricordiamo che un elemento  $x$  di un gruppo è *idempotente* se  $x^2 = x$ , cioè se  $x = x^2$ .

**Proposizione 4.** Ogni tavola di verità ad  $n$  variabili si può ottenere da una rete combinatoria (idempotente) reversibile ad  $n+1$  ingressi e quindi può essere simulata da una porta logica reversibile.

*Dim.* Sia  $t : \{0,1\}^n \rightarrow \{0,1\}$  una tavola di verità e definiamo  $t' : \{0,1\}^{n+1} \rightarrow \{0,1\}^{n+1}$  ponendo

$$t'(x_1, \dots, x_n, 1) = (x_1, \dots, x_n, t(x_1, \dots, x_n))$$

$$t'(x_1, \dots, x_n, 0) = (x_1, \dots, x_n, 1-t(x_1, \dots, x_n)).$$

Supponiamo che  $t'(x_1, \dots, x_n, z) = t'(x'_1, \dots, x'_n, z')$ , allora è evidente che  $x_1 = x'_1, \dots, x_n = x'_n$ . Non potendo risultare che  $t(x_1, \dots, x_n) = 1-t(x_1, \dots, x_n)$ ,  $z$  e  $z'$  sono entrambi uguali a 0 oppure

entrambi uguali ad 1. Pertanto abbiamo che  $z = z'$ . Ciò prova che  $t'$  è iniettiva. Per provare che  $t'$  è idempotente supponiamo che  $t(x_1, \dots, x_n) = 1$ , allora

$$\begin{aligned} t'(t'(x_1, \dots, x_n, 1)) &= t'(x_1, \dots, x_n, t(x_1, \dots, x_n)) = t'(x_1, \dots, x_n, 1) \\ &= (x_1, \dots, x_n, t(x_1, \dots, x_n)) = (x_1, \dots, x_n, 1). \end{aligned}$$

Inoltre

$$\begin{aligned} t'(t'(x_1, \dots, x_n, 0)) &= t'(x_1, \dots, x_n, 1-t(x_1, \dots, x_n)) = t'(x_1, \dots, x_n, 0) \\ &= (x_1, \dots, x_n, 1-t(x_1, \dots, x_n)) = (x_1, \dots, x_n, 0). \end{aligned}$$

In modo analogo si procede nel caso in cui  $t(x_1, \dots, x_n) = 0$ . Il fatto che  $t'$  sia idempotente comporta automaticamente che  $t'$  sia suriettiva. Infatti per ogni  $y \in \{0, 1\}^{n+1}$  risulta che  $t'(y)$  è un elemento tale che  $t'(t'(y)) = y$ .

Infine  $t(x_1, \dots, x_n)$  è la  $n+1$ -proiezione di  $t'(x_1, \dots, x_n, 1)$ .

Commento [d1]:

Ad esempio, per ottenere  $\wedge$  definiamo  $t'$  ponendo

$$\begin{aligned} t'(x, y, 1) &= (x, y, \min\{x, y\}) \\ t'(x, y, 0) &= (x, y, 1 - \min\{x, y\}). \end{aligned}$$

Possiamo ottenere il connettivo NAND ponendo

$$\begin{aligned} t'(x, y, 1) &= (x, y, 1 - \min\{x, y\}) \\ t'(x, y, 0) &= (x, y, \min\{x, y\}). \end{aligned}$$

Si ottiene la porta logica descritta dalla seguente tavola di verità

$p_1$	$p_2$	$p_3$	$O$	$O$	$C$
1	1	1	1	1	0
1	1	0	1	1	1
1	0	1	1	0	1
1	0	0	1	0	0
0	1	1	0	1	1
0	1	0	0	1	0
0	0	1	0	0	1
0	0	0	0	0	0

Da notare che tale porta logica, che prende il nome di *porta di Toffalori*, si ottiene scambiando i due elementi (1,1,1) e (1,1,0) che compaiono nelle prime due righe. Poiché il connettivo *NAND* è universale, nel senso che ogni tavola di verità può essere ottenuta tramite una rete di porte tutte uguali a quella di *NAND*, anche la porta logica di Toffalori è universale. Il teorema di completezza funzionale ci assicura che le tre porte logiche fondamentali sono sufficienti per potere realizzare ogni possibile tavola di verità. Nel caso di porte logiche reversibili la questione si pone in termini di sistemi di generatori di un gruppo.

**Definizione 5.** Un *sistema di generatori* di un gruppo  $G$  è un insieme  $X$  di elementi di  $G$  tale che ogni elemento di  $G$  è prodotto di elementi di  $X$  o di inversi di elementi di  $X$ .

La ricerca di un opportuno sistema di generatori  $f_1, \dots, f_h$  del gruppo  $G_{n+1}$  equivale a trovare un sistema di porte logiche capaci, insieme alle loro inverse, di generare per composizione ogni porta logica in  $G_{n+1}$  e quindi di realizzare ogni tavola di verità ad  $n$  ingressi.

### 11. Campi finiti e logiche a più valori per macchine più veloci (da finire).

Fino ad ora abbiamo considerato porte logiche che corrispondono alla logica classica in cui sono ammessi solo due valori di verità, 0 (falso) ed 1 (vero). Tuttavia esistono logiche in cui sono ammessi anche ulteriori valori di verità. Ad esempio può essere utilizzato il valore  $1/2$  per indicare che una asserzione contiene sia una certa dose di verità che di falsità. Più in generale, potrebbero essere utilizzati valori nell'insieme  $\{0/n, 1/n, 2/n, \dots, n/n\}$  per denotare  $n+1$  diversi livelli di verità fermo stando che  $0 = 0/n$  continua a denotare il falso e  $1 = n/n$  continua a denotare il vero. Tali logiche, che considereremo in maniera più dettagliata nel Capitolo 8, permetterebbero la costruzione di porte logiche e quindi di reti molto più veloci ed efficienti. Infatti se si individua un sistema di connettivi per una tale logica in modo che valga un teorema di completezza funzionale, allora costruendo porte logiche corrispondenti a avremmo reti lungo le cui linee di ingresso e di uscita possono viaggiare segnali di  $n+1$  diversi livelli di intensità. La completezza del sistema di connettivi assicurerebbe che sarebbe possibile simulare ogni possibile rete combinatoria ad  $n+1$  valori. Se si ammettono anche registri di memoria capaci di immagazzinare  $n+1$  livelli di segnale allora tale sistema di porte permetterebbe di costruire un qualunque automa finito (previa una opportuna codifica). Da un punto di vista teorico i sistemi informatici ottenuti in tale modo sarebbero notevolmente efficienti. Ad esempio se in 5 registri di memoria a valori 0 ed 1 possiamo scrivere (e quindi ricordare)  $2^5 = 32$  stringhe, se negli stessi registri possiamo scrivere dieci valori diversi allora possiamo scrivere  $10^5 = 100.000$  stringhe. In termini di trasmissione dell'informazione, basta pensare che per trasmettere una informazione codificata dal numero 101010 scritto in base 2, abbiamo bisogno di una sequenza di 6 segnali. Se invece lo stesso numero lo scriviamo in base dieci otteniamo 84 ed abbiamo bisogno di una sequenza di soli due segnali.

Come viene fatto nella logica con due valori, si pone il problema di definire il valore di verità di proposizione composte conoscendo il valore di verità delle componenti. Ad esempio deve essere stabilito quale valore di verità assegnare ad una asserzione del tipo "Mario è giovane e Mario è alto" conoscendo il valore di verità delle due componenti "Mario è giovane" e "Mario è alto". Questo significa che dobbiamo fare corrispondere ad ogni connettivo logico una operazione nell'insieme  $\{0, 1/2, 1\}$ . Ad esempio possiamo interpretare i connettivi  $\wedge, \vee$  con il minimo ed il massimo ottenendo le seguenti tavole di verità

$p_1$	$p_2$	$p_1 \vee p_2$	$p_1 \wedge p_2$
1	1	1	1
1	0.5	1	0.5
1	0	1	0
0.5	1	1	0.5
0.5	0.5	0.5	0.5
0.5	0	0.5	0
0	1	1	0
0	0.5	0.5	0
0	0	0	0

Possiamo inoltre valutare la negazione tramite la funzione  $1-x$  ottenendo la seguente tavola di verità

$p_1$	$\neg p_1$
1	0
0.5	0.5
0	1

Nella logica che ne esce fuori non vale né il principio del terzo escluso né il principio di non contraddizione. Ad esempio se la valutazione  $v$  è tale che  $v(p_1) = 0.5$

allora  $v(\neg p_1) = 1 - v(p_1) = 1 - 0.5 = 0.5$  e quindi

$$v(p_1 \wedge \neg p_1) = \min\{0.5, 0.5\} = 0.5 \quad ; \quad v(p_1 \vee \neg p_1) = \max\{0.5, 0.5\} = 0.5.$$

D'altra parte se qualcuno ci comunica che " $n$  è pari e dispari" noi pensiamo che stia dicendo qualche cosa di sicuramente sbagliato, se invece ci dice "Mario è alto e non alto" noi non pensiamo che stia dicendo una cosa falsa ma piuttosto che Mario non si possa collocare né tra le persone decisamente alte né tra quelle decisamente basse. Possiamo anche associare ad ogni formula  $\alpha$  le cui variabili proposizionali sono comprese tra  $p_1, \dots, p_n$  una *tavola di verità*  $t_\alpha: \{0, 1/2, 1\}^n \rightarrow \{0, 1/2, 1\}$  definita per ricorsione sulla complessità di  $\alpha$  tramite le equazioni

$$\begin{aligned} t_\alpha(x_1, \dots, x_n) &= x_i \quad \text{se } \alpha = p_i \\ t_{\alpha \wedge \beta}(x_1, \dots, x_n) &= \min\{t_\alpha(x_1, \dots, x_n), t_\beta(x_1, \dots, x_n)\} \\ t_{\alpha \vee \beta}(x_1, \dots, x_n) &= \max\{t_\alpha(x_1, \dots, x_n), t_\beta(x_1, \dots, x_n)\} \\ t_{\neg \alpha}(x_1, \dots, x_n) &= 1 - t_\alpha(x_1, \dots, x_n). \end{aligned}$$

**Teorema 1.** Il teorema di completezza funzionale non vale per la logica a tre valori che abbiamo proposto. Tuttavia aggiungendo la *negazione*  $\neg x$  definita ponendo

$$\neg 1 = 0, \quad \neg 0 = 1/2 \quad \text{and} \quad \neg 1/2 = 1$$

vale un teorema di completezza.

*Dim.* Osserviamo prima che se  $\alpha$  è una formula scritta con i connettivi  $\wedge, \vee, \neg$ , allora

$$t_\alpha(1/2, \dots, 1/2) = 1/2.$$

Ne segue che, ad esempio, la funzione costantemente uguale a 1 non è la tavola di verità di nessuna formula che si possa scrivere con tali connettivi. Omettiamo la dimostrazione della seconda parte del teorema.  $\square$

E' ragionevole introdurre ulteriori valori di verità. Ad esempio possiamo introdurre l'insieme di 10 valori di verità del tipo  $\{0, 1/9, 2/9, \dots, 9/9\}$  oppure, più in generale, l'insieme di  $n$  valori del tipo  $\{0, 1/(n-1), 2/(n-1), \dots, (n-1)/(n-1)\}$ . Anche in questo caso i connettivi  $\wedge, \vee, \neg$  possono essere valutati con il minimo, il massimo e la funzione  $1-x$ . Anche in questo caso tali connettivi non sono sufficienti per ottenere un teorema di completezza. Chiameremo *sistema completo di connettivi* per una logica ad  $n$  valori un insieme di connettivi per cui valga il teorema di completezza funzionale.

**Teorema 2.** Ogni logica ad un numero finito di valori di verità ammette un sistema completo di connettivi.

Purtroppo anche se promettente l'uso di porte logiche che lavorino con più di due livelli di segnale non è ancora possibile. Infatti la tecnologia relativa alla costruzione di porte logiche a più valori non è ancora abbastanza avanzata. Il problema è che la possibilità di errore di trasmissione dei segnali è notevolmente alta. Basta una leggera alterazione della tensione perché un valore del tipo  $5/9$  si trasformi in un valore del tipo  $6/9$  o  $4/9$ . D'altra parte la possibilità di errore è forte anche nei normali dispositivi digitali che lavorano con soli due valori. Per ovviare a tali pericoli usualmente viene usata una tecnica matematica che va sotto il nome di "codici correttori di errori".

**Cartesio<sup>7</sup>**

Dal *Discorso sul Metodo*: V parte (gli animali sono macchine ?):

Se ci fossero delle macchine le quali avessero gli organi e l'aspetto esterno della scimmia o di qualche animale irragionevole, noi non avremmo alcun mezzo per riconoscere che esse non fossero in tutto della stessa natura di quegli animali; laddove se ce ne fossero che avessero l'apparenza dei nostri corpi e imitassero le nostre azioni quanto sarebbe moralmente possibile, avremmo sempre due mezzi certissimi per riconoscere che esse non sarebbero punto perciò dei veri uomini. Dei quali il primo è che mai esse potrebbero adoperare parole o altri segni, componendoli come noi facciamo per dichiarare agli altri il nostro pensiero; perché si può ben concepire che una macchina sia costruita in tal guisa da proferire delle parole, e magari che ne proferisca alcune a proposito delle azioni corporee che cagioneranno qualche mutamento nei suoi organi, per esempio, che la si tocca in un dato posto, domandi che cosa le si può dire, e se un altro, che gridi che le si fa male, e simili; ma non già che esse le combinino in modo diverso per rispondere al senso di tutto ciò che si dirà in sua presenza, come al contrario possono fare i più ebeti tra gli uomini.

E il secondo, è che, quando pure esse facessero parecchie cose non meno bene e magari meglio di alcuni di noi, infallibilmente esse mancherebbero in certe altre cose, per cui si scoprirebbero che esse non agirebbero per conoscenza ma solo per la conformazione dei loro organi. Perché, laddove la ragione è uno strumento universale che può giovare in ogni sorta di occasione, quegli organi hanno bisogno di qualche particolare disposizione per ogni azione particolare; donde segue che è moralmente impossibile che ce ne siano assai e diversi in ogni macchina per farla agire in tutte le occorrenze della vita a quella guisa che agisce la nostra ragione.

Orbene, con questi due medesimi mezzi si può altresì conoscere la differenza che c'è tra gli uomini e le bestie. Perché è cosa notevolissima che non ci siano uomini così ebeti e stupidi, non eccettuati neppure gli insensati, che non siano in grado di combinare insieme diverse parole e comporre un discorso con cui facciano intendere i propri pensieri, e al contrario non c'è alcun altro animale perfetto e felicemente nato quanto si possa, che faccia niente di simile.

---

<sup>7</sup> Ricordo che Cartesio suddivideva la realtà in *res cogitans* e *res extensa*. Con la prima espressione intendeva la realtà psichica a cui Cartesio attribuisce le qualità: non avere estensione, libertà e consapevolezza. Invece la *res extensa* rappresenta la realtà fisica, che è, al contrario, estesa, limitata e inconsapevole. Egli riteneva che la prima fosse relativa solamente all'essere umano (e non agli animali). Molto più tardi, a partire da Turing, la ricerca scientifica si porrà un problema più difficile e si chiederà se l'uomo sia una macchina e che quindi non sia tutto riducibile alla sola *res extensa*.

Il che non accade perché abbiano deficienze di organi perché si vedono le gazze e pappagalli poter proferire le parole come noi, e non di meno non poter parlare come noi, ciò dando segni di non pensare a ciò che dicono laddove gli uomini, per essere nati sordi e muti, degli organi che servono agli altri per parlare sono privi altrettanto e più delle bestie, sogliono inventare essi stessi dei segni con cui farsi intendere da coloro che per trovarsi abitualmente con essi hanno modo di imparare il loro linguaggio.

Il che testimonia non soltanto che le bestie hanno meno di ragione che gli uomini ma piuttosto che esse non ne hanno affatto. Perché si vede che ne occorre ben poca per poter parlare e dato che si notano certe disuguaglianze tra gli animali della stessa specie come tra gli uomini, e gli uni siano più facili a educarsi che gli altri, non è credibile che una scimmia e un pappagallo tra i più perfetti della loro specie non potessero uguagliare un bambino tra i più stupidi, almeno un bambino dal cervello guasto, se la loro anima non fosse di una natura affatto differente dalla nostra.

Né sono da confondere le parole con i movimenti naturali rivelatori delle passioni che possono essere imitati come dagli animali; né è da pensare come alcuni antichi, che le bestie parlino sebbene noi non comprendiamo il loro linguaggio; perché se fosse vero, possedendo esse parecchi organi analoghi ai nostri, potrebbero farsi intendere da noi così bene come dai loro simili. E' altresì notevolissimo che, sebbene ci siano parecchi animali i quali rivelano maggiore industria di noi non dimostra già che abbiano dello spirito; perché a codesta stregua ne avrebbero più di ciascuno di noi e riuscirebbero meglio in ogni cosa: ma piuttosto che non ne hanno, e in loro agisce la natura secondo la disposizione dei loro organi; proprio come si vede che un orologio, composto semplicemente di ruote e di molle, può contare le ore e misurare il tempo meglio di noi con tutta la nostra saggezza.»