

LOGICA, INFORMAZIONE, AUTOMAZIONE

Introduzione per gli studenti dell'area umanistica

CLAUDIA CASADIO

Logica e Filosofia della Scienza
SSD M-FIL/02

claudia.casadio@unich.it

Università degli Studi di Chieti e Pescara
Dipartimento di Lingue, Letterature e Culture Moderne
Viale Pindaro 42, 65127 Pescara

INDICE

PARTE I - LOGICA E APPLICAZIONI

CAPITOLO I - BASI LOGICHE

- 1. Concetti logici**
 - 1.1 *Proposizioni*
 - 1.2 *Connettivi logici*
 - 1.3 *Proprietà logiche*

 - 2. Operatori booleani**
 - 2.1 *Connettivi logici e circuiti*
 - 2.2 *Operatori booleani e dualità*
 - 2.3 *Porte logiche*

 - 3. Argomenti e inferenze logiche**
 - 3.1 *Inferenze e organizzazione delle conoscenze*
 - 3.2 *Dimostrazioni vs. refutazioni*
 - 3.3 *Dualità e informazione*
- ESERCIZI

CAPITOLO II - ONTOLOGIE

- 1. Nozioni preliminari**
 - 2. Categorizzazione e percezione**
 - 2.1 *Albero di Porfirio*
 - 2.2 *Teoria Classica*
 - 2.3 *Teoria del Prototipo*
 - 2.4 *Teoria Funzionalista*
 - 3. Rappresentatività**
 - 3.1 *Modelli Rappresentazionali*
 - 3.2 *Reti Semantiche*
 - 4. Organizzazione delle conoscenze tramite ontologie**
- ESERCIZI

PARTE II - INFORMAZIONE E AUTOMAZIONE

CAPITOLO III - INFORMAZIONE

- 1. Trattamento simbolico delle informazioni**
 - 1.1 *Cenni storici*
 - 1.2 *Algoritmi*
 - 2. Costruzione di grafi per algoritmi**
 - 2.1 *Diagrammi di flusso*
 - 2.2 *Esempi di diagrammi di flusso*
 - 2.3 *Ricorsività*
- ESERCIZI

CAPITOLO IV - AUTOMAZIONE

- 1. Cenni sulla nascita della cibernetica**
 - 1.1 *Principi teorici*
 - 1.2 *Cibernetica e comunicazione*
 - 2. Intelligenza artificiale**
 - 2.1 *Sistemi esperti*
 - 3. Macchine e automi**
 - 3.1 *Strategie*
 - 3.2 *Macchine*
 - 3.3 *Macchina di Turing e calcolabilità*
 - 3.3.1 *Turing e l'idea di Intelligenza Artificiale*
 - 3.3.2 *Macchina di Turing*
 - 4. Esperimenti mentali**
 - 4.1 *Test di Turing*
 - 4.2 *Stanza cinese di Searle*
 - 5. Automi**
 - 5.1 *Automi e linguaggi*
 - 5.2 *Automi a stati finiti*
 - 5.3 *Automi a pila*
 - 5.4 *Automi non deterministici*
 - 6. Automi, linguaggi, grammatiche**
 - 6.1 *Gerarchia di Chomsky*
- ESERCIZI

BIBLIOGRAFIA

INDICE ANALITICO

PARTE I

LOGICA E APPLICAZIONI

CAPITOLO PRIMO

BASI LOGICHE

1. CONCETTI LOGICI

In via preliminare è opportuno richiamare i fondamentali concetti logici che saranno di riferimento per il nostro lavoro: proposizione, connettivo, tautologia, contraddizione, dimostrazione¹.

1.1 PROPOSIZIONI

In ogni area della conoscenza e della comunicazione ci sono proposizioni con le quali vengono espresse informazioni, sono trasmessi dati, notizie, sono sviluppati ragionamenti e argomentazioni. Le proposizioni sono dette anche *enunciati*, *asserzioni*, *affermazioni* e possono essere semplici o complesse; possono contenere componenti che esprimono concetti logici, in particolare gli operatori e i connettivi che consentono di formare proposizioni complesse a partire da proposizioni semplici. Le proposizioni sono studiate per il loro *significato* (semantica), il loro *uso* (pragmatica), per le loro proprietà *strutturali* (sintassi) e le loro proprietà *logiche* (forma logica)².

Proposizioni semplici:

Il gatto è un felino. Le città sono affollate. Il numero 3 è dispari.

Proposizioni complesse:

Il gatto è un felino e 3 è un numero dispari.

Se corro, mi muovo. Mi muovo se e solo non sto fermo.

Quando arriva l'autunno e subentra l'ora solare, i giorni si accorciano.

Le proposizioni studiate in primo luogo dalla logica sono gli enunciati *dichiarativi* al modo indicativo, in genere al tempo presente o passato e alla terza persona singolare o plurale, come negli esempi seguenti:

Venere è un pianeta.

L'acqua gela a 0 gradi centigradi.

Il numero 3 è dispari e il numero 8 è la somma di 6 e 2.

¹ Questi concetti sono trattati in dettaglio nel manuale *Logica e psicologia del pensiero* (Casadio 2006); si vedano anche Abrusci (2009), Palladino (2002).

² *Sintassi, semantica e pragmatica* rappresentano i tre livelli fondamentali in cui sono organizzati i sistemi di analisi teorica dei linguaggi sia formali che naturali; si veda Bazzanella (2005).

Non ricadono in questa categoria gli enunciati dotati di vari tipi di forza illocutoria³, o contenenti elementi deittici ed indicali:

Mi hai chiamato?
 Alzate tutti la mano sinistra!
 Dubito che tu possa capire quello che dico.
 Domani ci sarà un temporale.

Pur essendo suscettibili di una analisi formale e di rappresentazione semantica, come è stato dimostrato dagli sviluppi della analisi logica del linguaggio naturale⁴, enunciati come *domande*, *comandi*, *esclamazioni*, oppure enunciati *temporali* ed *indicali*, diversamente dalle proposizioni dichiarative, non possono essere soltanto *asseriti* o *negati*. Solo le proposizioni asseriscono o negano stati di fatto e perciò possono essere VERE o FALSE.

1.2 CONNETTIVI LOGICI

Nel calcolo delle proposizioni o logica proposizionale sono definiti i seguenti cinque operatori o connettivi *logici* fondamentali:

NEGAZIONE : $\sim A$

Si legge “non A”, si interpreta “non si dà il caso che A”.

CONGIUNZIONE: $A \wedge B$

Si legge “e”, si interpreta “la proposizione A e la proposizione B valgono insieme”.

DISGIUNZIONE: $A \vee B$

Si legge “o”, si interpreta “vale la proposizione A o la proposizione B”, in senso inclusivo.

CONDIZIONALE o IMPLICAZIONE: $A \rightarrow B$

Si legge “se A allora B”, dove A è l’antecedente. B il conseguente del condizionale; si interpreta “se vale la proposizione A, allora vale anche la proposizione B”, oppure, “la proposizione B dipende dalla proposizione A”.

EQUIVALENZA o BICONDIZIONALE: $A \leftrightarrow B$

Si legge “A se e solo se B”; si interpreta “la proposizione A e la proposizione B si equivalgono”, ovvero, “non si dà il caso che valga A senza che valga B”.

³ La distinzione tra forza *locutoria* e forza *illocutoria* di un enunciato è stata formulata da Austin (1962) ed è alla base della teoria degli atti linguistici. Mentre la prima rappresenta la semplice emissione del contenuto enunciativo, la seconda esprime le varie modalità assunte dal proferimento; si vedano Picardi (1992), (1999), Bazzanella (2005).

⁴ Si vedano, ad esempio, Chierchia (1997), Marraffa (2003), Croft e Cruse (2010).

Questi connettivi sono detti logici perché introducono operazioni *vero-funzionali*: la verità o la falsità di una proposizione complessa è determinata dalla verità o falsità delle proposizioni che la formano in base alle connessioni istituite dai connettivi. Il significato di ciascun connettivo è espresso da una funzione di verità i cui valori sono $\{0, 1\}$ (valori di verità o valori booleani: 0 = FALSO, 1 = VERO) e il cui andamento è mostrato dalle tavole di verità della logica proposizionale. Le seguente tabella riassume le proprietà dei cinque connettivi proposizionali considerati.

DEFINIZIONE	NOTAZIONE	SIGNIFICATO
<i>Negazione</i>	\sim oppure \neg	non, non si dà il caso che, <i>not</i>
<i>Congiunzione</i>	\wedge oppure \bullet	e, <i>et, and</i>
<i>Disgiunzione (inclusiva)</i>	\vee	o, <i>vel, or</i>
<i>Implicazione o condizionale</i>	\rightarrow oppure \supset	se ... allora, implica
<i>Equivalenza o Bi condizionale</i>	\leftrightarrow oppure \equiv	se e solo se, equivale

TABELLA 1. *Connettivi proposizionali*

Nella logica proposizionale classica abbiamo cinque funzioni di verità che corrispondono a questi connettivi, come mostrato nelle seguenti tabelle.

NEGAZIONE: se A è una proposizione, $\sim A$ è la sua negazione, con la condizione di verità: se A è VERA (valore 1), $\sim A$ è FALSA (valore 0); se A è FALSA (valore 0), $\sim A$ è VERA (valore 1).

A	$\sim A$
1	0
0	1

TABELLA 2. *Negazione*

CONGIUNZIONE: date le proposizioni A , B , la proposizione $A \wedge B$ è la loro congiunzione, con la condizione di verità: $A \wedge B$ è VERA (valore 1) se e solo se *sia* A *che* B sono VERE; se o A , o B o entrambe sono FALSE, allora $A \wedge B$ è FALSA (valore 0).

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

TABELLA 3. *Congiunzione*

DISGIUNZIONE (*inclusiva*): date le proposizioni A, B, la proposizione $A \vee B$ è la loro disgiunzione, con la condizione di verità: $A \vee B$ è VERA (valore 1) se o A è VERA, o B è VERA o entrambe sono VERE; $A \vee B$ è FALSA (valore 0) se e solo se *sia A che B* sono FALSE.

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

TABELLA 4. *Disgiunzione*

IMPLICAZIONE (*condizionale materiale*): date le proposizioni A, B, la proposizione $A \rightarrow B$ è il condizionale A *implica* B (se A *allora* B), in base alla condizione di verità: $A \rightarrow B$ è VERA (valore 1) se e solo se *non si dà il caso che* l'antecedente A sia VERO (valore 1) e il conseguente B FALSO (valore 0); $A \rightarrow B$ è FALSA (valore 0) se A è VERA e B è FALSA.

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

TABELLA 5. *Condizionale*

EQUIVALENZA (*bicondizionale*): date le proposizioni A, B, l'equivalenza $A \leftrightarrow B$ (A *se e solo se* B) è una proposizione, con la condizione di verità: $A \leftrightarrow B$ è VERA (valore 1) se e solo se A e B sono entrambe VERE e, o entrambe FALSE; altrimenti $A \leftrightarrow B$ è FALSA (valore 0).

A	B	$A \leftrightarrow B$
1	1	1
1	0	0
0	1	0
0	0	1

TABELLA 6. *Bicondizionale*

1.3 PROPRIETÀ LOGICHE

Nella logica proposizionale o, calcolo delle proposizioni, le proposizioni complesse (molecole) sono ottenute collegando proposizioni semplici (atomi) mediante i connettivi logici, in particolare quelli sopra menzionati. Ad ogni proposizione così formata è associata una corrispondente funzione di verità, i cui valori sono $\{0, 1\}$, che risulta dalla composizione delle funzioni di verità dei connettivi che occorrono nella proposizione⁵; quando la funzione di verità determinata da una proposizione assume sempre il valore 1 (VERO) si dice che la proposizione è una *tautologia* del calcolo delle proposizioni; quando la funzione di verità assume sempre il valore 0 (FALSO), abbiamo una *contraddizione*; infine, quando la funzione di verità assume in certi casi il valore 1 (VERO) ed in certi casi il valore 0 (FALSO), si dice che la proposizione è *contingente*. Esprimiamo questi concetti con le seguenti definizioni:

TAUTOLOGIA

Una proposizione è una *tautologia* se e solo se la funzione di verità da essa determinata assume sempre il valore 1 (VERO) per qualsiasi assegnazione di valori di verità alle sue parti costituenti (variabili proposizionali). Sono tautologie le proposizioni logiche⁶: $A \vee \sim A$, $\sim(A \wedge \sim A)$, $\sim(\sim A) \leftrightarrow A$.

CONTRADDIZIONE

Una proposizione è una *contraddizione* se e solo se la funzione di verità da essa determinata assume sempre il valore 0 (FALSO) per qualsiasi assegnazione di valori di verità alle sue parti costituenti (variabili proposizionali). Sono contraddizioni le proposizioni: $A \wedge \sim A$, $\sim A \leftrightarrow A$.

CONTINGENZA

Una proposizione si dice *contingente* se e solo se la funzione di verità da essa determinata assume il valore 1 (VERO) per certe assegnazioni di valori di verità alle sue parti costituenti (variabili proposizionali) e il valore 0 (FALSO) per certe altre assegnazioni di valori di verità alle sue parti costituenti (variabili proposizionali). Sono contingenti le proposizioni: $A \wedge B$, $\sim A \vee B$, $A \leftrightarrow B$.

⁵ Il procedimento più noto a cui si fa ricorso per determinare se una proposizione è una tautologia (o una contraddizione) è il metodo delle TAVOLE DI VERITÀ; per una illustrazione ed esempi si vedano Casadio (2006, pp. 51-60), Abrusci (2010, pp. 71-85).

⁶ Si tratta, rispettivamente delle leggi logiche: *Terzo escluso*, *Principio di non contraddizione*, *Legge della doppia negazione*.

Una tautologia è una proposizione che risulta vera per qualunque assegnazione di valori di verità, dunque indipendentemente dalla situazione di fatto. Per questa ragione le tautologie sono dette anche *leggi logiche*. Gli esempi di tautologie che abbiamo indicato sono *leggi logiche* del calcolo delle proposizioni. D'altro lato, una contraddizione è una proposizione che risulta sempre falsa, per qualunque assegnazione di valori di verità, indipendentemente dalla situazione di fatto; in particolare, la *negazione* di una tautologia è sempre una contraddizione (e viceversa).

Possiamo interpretare i valori 1 e 0 come le due posizioni di un interruttore di corrente elettrica, in cui il valore 1 indica la situazione in cui l'interruttore è chiuso e la corrente passa, mentre il valore 0 indica la situazione in cui l'interruttore è aperto e la corrente non passa (vedi fig. 1). I connettivi logici possono essere interpretati in questo modo e, in particolare, una tautologia corrisponde alla situazione in cui si ha sempre passaggio di corrente, per qualsiasi configurazione delle sue connessioni, mentre una contraddizione corrisponde alla situazione in cui la corrente non può passare mai.

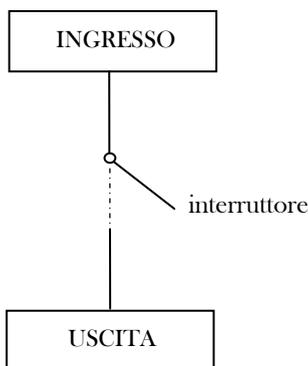


FIGURA 1

2. OPERATORI BOOLEANI

L'algebra di Boole⁷, o algebra booleana, è un ramo dell'algebra astratta che comprende tutte le algebre che operano con i soli valori $\{0, 1\}$, detti variabili

⁷ George Boole (1815-1864) è uno dei fondatori dell'algebra astratta e in particolare dell'algebra della logica o algebra di Boole: una struttura algebrica consistente di un insieme di oggetti su cui sono definite due operazioni binarie: \times (prodotto), $+$ (somma) e una operazione unaria $'$ (complemento). L'esempio più noto di algebra di Boole, fondamentale in informatica, è l'algebra dei valori 1 (vero), 0 (falso) per i connettivi logici classici \wedge (*and*), \vee (*or*), \neg (*not*).

booleane o logiche. La struttura algebrica studiata dall'algebra booleana è finalizzata all'analisi logica delle proposizioni nell'ambito del calcolo proposizionale.

L'algebra booleana definisce gli operatori logici AND (connettivo \wedge), OR (connettivo \vee), NOT (connettivo \neg), la cui combinazione permette di sviluppare qualsiasi funzione logica e consente di trattare in termini esclusivamente algebrici le operazioni logiche di *coniunzione*, *disgiunzione* (inclusiva) e *negazione*, corrispondenti alle operazioni insiemistiche di *intersezione*, *unione* e *complemento*, e le altre operazioni definibili sulla base di queste.

Mentre l'algebra elementare si basa sulle operazioni numeriche di *prodotto* o *moltiplicazione*: $x \cdot y$, *somma* o *addizione* $x + y$, e *differenza* $x - y$, l'algebra booleana fa riferimento alle controparti logiche di quelle operazioni, la *coniunzione* $x \wedge y$ (AND), la *disgiunzione* $x \vee y$ (OR), il *complemento* o negazione $\neg x$ (NOT). Come l'operazione logica di *coniunzione*, la connessione AND di due proposizioni è VERA (valore 1) quando entrambe le proposizioni sono VERE, altrimenti è FALSA (valore 0) (vedi TABELLA 3). Come l'operazione logica di *disgiunzione*, la connessione OR di due proposizioni è VERA (valore 1) quando entrambe le proposizioni sono VERE, o una delle due è VERA, altrimenti è FALSA (valore 0) (vedi TABELLA 4).

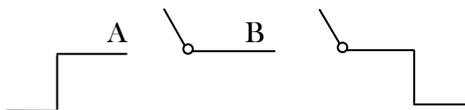
La negazione logica, invece, non corrisponde esattamente alla differenza numerica. Con l'operazione numerica condivide la proprietà che applicandola due volte si ritorna al valore originale: $\neg(\neg x) = x$. Questa proprietà dell'operatore NOT è detta *involutione* e ha come corrispondente la legge logica classica detta della *doppia negazione*: $\sim(\sim A) \leftrightarrow A$.

2.1 CONNETTIVI LOGICI E CIRCUITI

Un circuito elettrico che contiene solo interruttori a due posizioni, così che quando un interruttore è chiuso passa la corrente (valore 1), quando è aperto la corrente non passa (valore 0), può essere rappresentato da un diagramma in cui ciascun interruttore è rappresentato da una lettera, che corrisponde ad un enunciato del calcolo proposizionale⁸.

⁸ Gli studi nel campo dell'elettronica digitale e dei circuiti elettrici (*digital computing, digital circuits*) ed i collegamenti con l'algebra booleana e le operazioni della logica si devono in primo luogo a Claude Shannon (1916-2001) divenuto celebre per avere fondato la teoria dell'informazione con un lavoro seminale pubblicato nel 1948. Ma già nel 1937, al MIT, scrisse una tesi sulle applicazioni dell'algebra booleana e della logica proposizionale all'elettronica. Si vedano Shannon (1938),(1948), Shannon e Weaver (1939).

In particolare, la forma proposizionale *coniuntiva* $A \wedge B$ è rappresentata da un circuito con due interruttori in serie (circuito *seriale*); la corrente passa solo se entrambi gli interruttori sono chiusi (entrambe le proposizioni sono vere); l'immagine mostra la situazione in cui entrambi gli interruttori sono aperti (la proposizione congiuntiva è falsa)



La seguente tabella mostra come viene rappresentata la funzione di verità della congiunzione nei termini delle possibili posizioni assunte da tale circuito

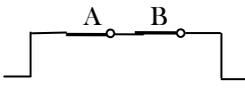
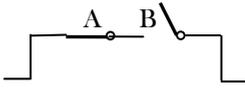
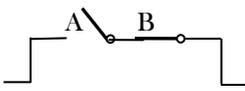
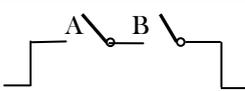
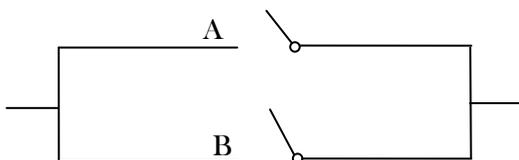
	A	B	$A \wedge B$
	1	1	1
	1	0	0
	0	1	0
	0	0	0

TABELLA 7. Circuito della congiunzione

La forma proposizionale *disgiuntiva* $A \vee B$ è rappresentata da un circuito con due interruttori in parallelo, la corrente passa quando uno dei due interruttori è chiuso o entrambi sono chiusi; l'immagine mostra la situazione in cui entrambi gli interruttori sono aperti (la proposizione disgiuntiva è falsa)



La tabella mostra come viene rappresentata la funzione di verità della disgiunzione nei termini delle possibili posizioni assunte da tale circuito

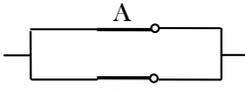
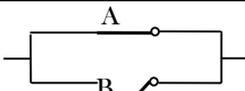
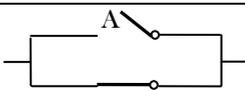
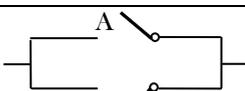
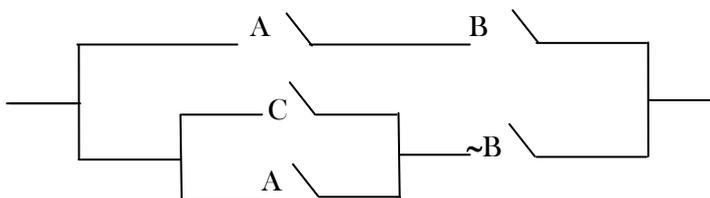
	A	B	$A \vee B$
	1	1	1
	1	0	1
	0	1	1
	0	0	0

TABELLA 8. *Circuito della disgiunzione*

Ovviamente i circuiti si possono combinare tra loro come mostra questo diagramma che rappresenta la forma proposizionale: $(A \wedge B) \vee ((C \vee A) \wedge \sim B)$



La relazione tra le proprietà delle operazioni booleane, le corrispondenti operazioni logiche e la loro rappresentazione in base a circuiti elettrici, o alternativamente, la perfetta rappresentatività dei circuiti elettrici in base a tali operazioni, è stata chiaramente delineata da Claude Shannon, di cui riportiamo il seguente brano⁹:

⁹ Shannon (1938, p. 471).

Nei circuiti di gestione e di controllo dei sistemi elettrici è spesso necessario fare intricate connessioni di interruttori e relè, come avviene nelle connessioni telefoniche o nella maggior parte dei circuiti realizzati per eseguire operazioni complesse in modo automatico. È necessario formulare una analisi matematica delle proprietà di questi reticoli, prestando particolare attenzione al problema della sintesi del reticolo. Il metodo può essere descritto brevemente nel modo seguente: un circuito è rappresentato da un insieme di equazioni, i cui termini corrispondono ai vari relè e interruttori del circuito. Viene sviluppato un calcolo per manipolare queste equazioni in base a semplici procedimenti matematici, che sono nella maggior parte dei casi simili a ordinari algoritmi algebrici. Questo calcolo è esattamente analogo al calcolo delle proposizioni usato nello studio simbolico della logica. Per il problema della sintesi le caratteristiche desiderate vengono prima scritte come un sistema di equazioni, poi le equazioni sono manipolate nella forma che rappresenta il circuito più semplice. Tale circuito può quindi essere immediatamente disegnato dalle equazioni. In base a questo metodo è sempre possibile trovare il più semplice circuito che contiene solo connessioni *seriali* e *parallele*, e in certi casi il circuito più semplice che contiene entrambe le connessioni.

2.2 OPERATORI BOOLEANI E DUALITÀ

Gli operatori booleani AND e OR, analogamente i connettivi della logica proposizionale \wedge (congiunzione) e \vee (disgiunzione), sono caratterizzati dalla fondamentale proprietà della reciproca *dualità*, ottenuta attraverso l'applicazione dell'operatore NOT (la negazione logica \sim). Nella logica proposizionale questa proprietà viene espressa dalle seguenti leggi (o teoremi), conosciute come Leggi di De Morgan¹⁰:

$$\text{DM1 } \sim(A \wedge B) \leftrightarrow \sim A \vee \sim B$$

$$\text{DM2 } \sim(A \vee B) \leftrightarrow \sim A \wedge \sim B$$

che nella formulazione equivalente

$$A \wedge B \leftrightarrow \sim(\sim A \vee \sim B)$$

$$A \vee B \leftrightarrow \sim(\sim A \wedge \sim B)$$

consentono di definire la congiunzione logica \wedge nei termini della disgiunzione \vee , e vice versa. Intuitivamente, il significato della prima legge DM1 è che la

¹⁰ Augustus De Morgan (1806-1871), matematico e logico britannico; sviluppò lo studio dell'algebra astratta, in particolare dell'algebra delle relazioni, che fu poi estesa da Charles Peirce, Ernst Schröder e ripresa successivamente nei *Principia Mathematica* di Bertrand Russell e Alfred N. Whitehead (Whitehead e Russell 1910/13).

negazione di una forma congiuntiva, in cui occorrono due proposizioni A e B , equivale ad una forma disgiuntiva in cui entrambe A e B occorrono negate; analogamente, il significato della seconda legge DM2 è che la negazione di una forma disgiuntiva, in cui occorrono due proposizioni A e B , equivale ad una forma congiuntiva in cui entrambe A e B occorrono negate.

Nei termini dell'algebra booleana la prima legge può essere così formulata usando la notazione di David Hilbert¹¹, nella quale la negazione viene indicata con un tratto tracciato sopra l'espressione da negare

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

Il significato è che, date due classi (o insiemi) A , B , se un elemento non appartiene al prodotto (intersezione) $A \cdot B$, allora o non appartiene ad A , o non appartiene a B o non appartiene ad entrambi. La seconda legge viene così formulata

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

con l'interpretazione: date due classi (o insiemi) A , B , se un elemento non appartiene alla somma (unione) $A + B$, allora non appartiene ad A e non appartiene a B . La stessa notazione può essere applicata alla formulazione della leggi De Morgan per la logica proposizionale, dove A e B sono proposizioni. La notazione di Hilbert si applica in base ad una semplice procedura: (i) si scrive l'espressione con la linea tracciata sopra; (ii) la linea viene spezzata in mezzo; (iii) l'operatore viene capovolto, p.es. la congiunzione \wedge diventa la disgiunzione \vee ; ad esempio

$$(i) \overline{p \wedge q} ; (ii) \overline{p} \wedge \overline{q} ; (iii) \overline{p} \vee \overline{q}$$

Questi due teoremi possono essere estesi ad un numero grande a piacere di proposizioni (classi, insiemi), ad esempio

$$\overline{A \cdot B \cdot C \cdot D \cdot \dots} = \overline{A} + \overline{B} + \overline{C} + \overline{D} + \dots$$

¹¹ David Hilbert (1862-1943) logico e matematico tedesco, ha contribuito con la sua opera alla fondazione della logica formale contemporanea, in particolare con le sue ricerche sulla assiomatizzazione della geometria.

2.3 PORTE LOGICHE

Mostriamo qui sotto i simboli usati in elettronica digitale¹² per rappresentare gli operatori NOT, AND, OR e altri importanti operatori basati su di essi: le porte di ingresso, o *input*, sono a sinistra e il segnale passa attraverso le porte di uscita, o *output*, sulla destra. I circoli rappresentano l'inversione (negazione) dei segnali di input o di output.

OPERATORE NOT

Come la negazione logica (vedi TABELLA 2), l'operatore NOT restituisce il valore inverso rispetto a quello in entrata¹³. Il simbolo di una porta NOT è



Spesso, al fine di semplificare espressioni complesse, si usano operatori che uniscono l'operazione NOT ad altre: questi operatori sono NOR (OR + NOT), NAND (AND + NOT), XNOR (XOR + NOT). La negazione, in questi casi, viene applicata dopo il risultato dell'operatore principale (OR, AND, XOR).

OPERATORE OR

L'operazione logica OR restituisce il valore 1 se almeno uno degli elementi (costituenti, proposizioni) ha valore 1, mentre restituisce 0 in tutti gli altri casi (vedi TABELLA 4). Tale operazione è anche detta somma logica. Nella teoria degli insiemi corrisponde all'*unione*. Il simbolo di una porta OR è:



¹² In informatica la parola *bit* può significare: (i) unità di misura dell'informazione, unità minima di memorizzazione, (dall'inglese *binary unit*), definita come la quantità minima di informazione che serve a distinguere tra due possibili alternative; (ii) una cifra binaria, (in inglese *binary digit*), uno dei due simboli del sistema numerico binario, *zero* (0) e *uno* (1); una metafora molto usata è quella della lampadina accesa o spenta. Il *bit* di per sé non dice nulla di comprensibile, solo raggruppandoli (normalmente 8 alla volta) otteniamo una entità significativa, chiamata *byte*.

¹³ Una concatenazione di NOT è semplificabile con un solo NOT in caso di occorrenze dispari o con nessuno nel caso di occorrenze pari.

OPERATORE AND

L'operazione AND dà come valore 1 se tutti gli elementi (costituenti, proposizioni) hanno valore 1, mentre restituisce il valore 0 in tutti gli altri casi (vedi TABELLA 3). Tale operazione è anche detta prodotto logico e nella teoria degli insiemi corrisponde all'*intersezione*. Il simbolo di una porta and è:



OPERATORE XOR

L'operatore XOR, detto anche EX-OR o OR *esclusivo*, è introdotto dal simbolo ' \oplus ': $A \oplus B$ restituisce il valore 1 *se e solo* se uno solo dei due elementi (proposizioni) A o B ha valore 1, mentre restituisce il valore 0 in tutti gli altri casi, come mostra la tabella

A	B	$A \oplus B$
1	1	0
1	0	1
0	1	1
0	0	0

TABELLA 9. *Disgiunzione esclusiva*

Il simbolo di una porta XOR è:



OPERATORE NAND

L'operatore NAND, che è la negazione del risultato dell'operazione AND, è introdotto dal simbolo ' \uparrow '¹⁴; $A \uparrow B$ restituisce il valore 0 se e solo se tutti gli elementi (proposizioni) hanno valore 1, mentre restituisce il valore 1 in tutti gli altri casi, come mostra la tabella

¹⁴ Questo simbolo è noto come '*stroke*' di Sheffer.

A	B	$A \uparrow B$
1	1	0
1	0	1
0	1	1
0	0	1

TABELLA 10. Operatore NAND

Il simbolo di una porta NAND , composta da un operatore NOT in serie con un operatore AND, è:



OPERATORE NOR

L'operatore NOR, che è la negazione del risultato dell'operazione OR, è introdotto dal simbolo ' \downarrow '; $A \downarrow B$ restituisce il valore 1 se e solo se tutti gli elementi (proposizioni) hanno valore 0, mentre restituisce il valore 0 in tutti gli altri casi, come mostra la tabella

A	B	$A \downarrow B$
1	1	0
1	0	0
0	1	0
0	0	1

TABELLA 11. Operatore NOR

Il simbolo di una porta NOR, composta da un operatore NOT in serie con un operatore OR, è:



OPERATORE XNOR

L'operatore XNOR, detto anche EX-NOR, è la negazione del risultato dell'operazione XOR; detto anche equivalenza logica, restituisce il valore 1 se tutti gli elementi (proposizioni) hanno valore 1, oppure se tutti gli elementi hanno valore 0, come mostra la tabella (cf. tabella 6)

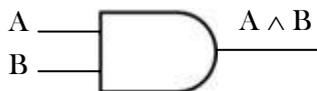
A	B	A XNOR B
1	1	1
1	0	0
0	1	0
0	0	1

TABELLA 11. Operatore XNOR

Il simbolo di una porta XNOR è:



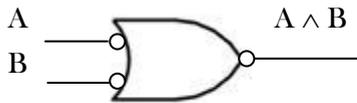
Vediamo ora alcune esemplificazioni di proprietà logiche mediante gli operatori e le porte che abbiamo definito. La congiunzione logica di due proposizioni è data dalla porta AND con due ingressi A, B e un'uscita $A \wedge B$



La disgiunzione inclusiva di due proposizioni è data dalla porta OR con due ingressi A, B e un'uscita $A \vee B$



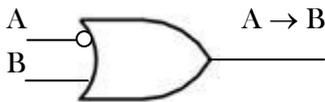
Con le porte NAND e NOR possiamo rappresentare le due leggi De Morgan nel modo seguente: l'equivalenza $\sim(\sim A \vee \sim B) \leftrightarrow A \wedge B$ è data dalla porta NOR



mentre l'equivalenza $\sim(\sim A \wedge \sim B) \leftrightarrow A \vee B$ è data dalla porta NAND



Nel calcolo delle proposizioni un enunciato condizionale è introdotto dal connettivo logico ' \rightarrow ', che si legge 'se A allora B', dove la proposizione A è l'*antecedente* e la *proposizione* B il conseguente del condizionale; nei termini dell'equivalenza logica¹⁵ $(A \rightarrow B) \leftrightarrow (\sim A \vee B)$, un enunciato condizionale è dato dalla porta OR



mentre la sua duale De Morgan $\sim(A \wedge \sim B) \leftrightarrow (A \rightarrow B)$ è data dalla porta AND



Per concludere, mostriamo nella tabella che segue, in cui sono riportati i relativi valori di verità, la relazione che intercorre tra gli operatori NAND vs. OR e NOR vs. AND, in coerenza con le leggi De Morgan; nella prima parte si vede come l'applicazione dell'operatore NAND ad A e B corrisponde alla applicazione dell'operatore OR alle negazioni $\sim A$ e $\sim B$; nella seconda parte della tabella si vede come l'applicazione dell'operatore NOR alle negazioni $\sim A$ e $\sim B$ corrisponde alla applicazione dell'operatore AND ad A e B.

¹⁵ Si veda Casadio (2005, p. 59).

A	B	A AND B	A NAND B = ($\sim A$ OR $\sim B$)
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	1
$\sim A$	$\sim B$	$\sim A$ OR $\sim B$	$\sim A$ NOR $\sim B = (A$ AND $B)$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

TABELLA 12.

2.4 FUMO O NON FUMO, COSA MI SERVE?

Le porte logiche possono essere impiegate per rappresentare l'organizzazione e la combinazione di vari tipi di risorse o informazioni e le relazioni che intercorrono tra di esse. Un esempio, che impiega un insieme di porte AND e OR è il seguente. Ho a disposizione le seguenti risorse e informazioni:

- S Sigaretta
- P Pipa
- T Tabacco
- F Fiammiferi
- A Accendino
- VF Voglia di Fumare
- DF Divieto di Fumare

e costruisco un digramma delle condizioni in base a cui si determina se posso o non posso fumare; un possibile diagramma di questo tipo è generato dalla sequenza di istruzioni (rappresentata nella Fig. 2):

$$((P \text{ AND } T) \text{ OR } S) \text{ AND } (F \text{ OR } A) \text{ AND } VF) \text{ AND } (\text{NOT } DF)$$

che corrisponde alla seguente proposizione logica, in cui le variabili proposizionali sono le iniziali dei nomi delle risorse e delle istruzioni:

$$((P \wedge T) \vee S) \wedge (F \vee A) \wedge VF) \wedge (\sim DF) .$$

La sequenza di istruzioni viene eseguita dal programma:

IF (((Pipa **AND** Tabacco) **OR** Sigaretta) **AND** (Fiammiferi **OR** Accendino)) **AND**
Voglia-Di-Fumare) **AND** **NOT** Divieto-Di-Fumare) **THEN**
 Messaggio ("SI posso fumare")
Else
 Messaggio ("NON posso fumare")
End IF

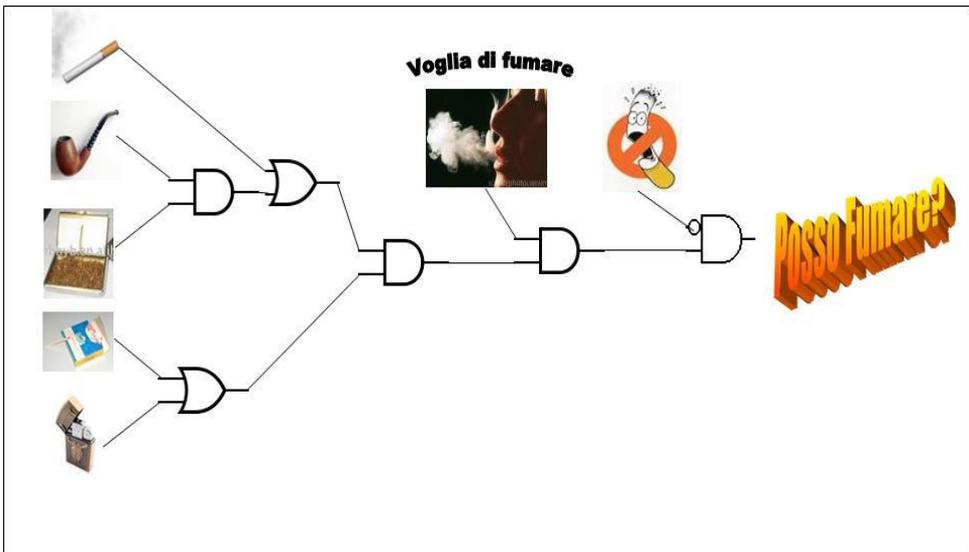


FIGURA 2. Diagramma: posso fumare?

3. ARGOMENTI E INFERENZE LOGICHE

3.1 INFERENZE E ORGANIZZAZIONE DELLE CONOSCENZE

Gli argomenti o argomentazioni sono insiemi di proposizioni collegate tra loro da qualche proprietà, che può essere di natura semantica, pragmatica o, in particolare, logica. Ad esempio, i seguenti insiemi di proposizioni sono argomenti

- | | |
|------------------------------------|--|
| 1. Tutti i felini hanno i polmoni. | 1. Quando viene l'autunno le giornate si accorciano. |
| 2. I gatti sono felini. | 2. Quindi bisogna accendere la luce presto la sera. |
| 3. I gatti hanno i polmoni | |

Mente non lo sono i seguenti

- | | |
|--|---------------------------------------|
| 1. Ho visitato molte città d'arte questa estate. | 1. I numeri reali sono infiniti. |
| 2. Mi è anche piaciuto fare il bagno al mare. | 2. I numeri pari non sono dispari. |
| 3. Ho conosciuto molte persone interessanti. | 3. I numeri primi non hanno divisori. |

Argomenti logici sono quelli in cui le proposizioni che vi compaiono sono collegate tra loro da proprietà logiche. Questi argomenti sono dotati di una certa organizzazione, o struttura, formale detta *forma logica*. In particolare, *sono logiche quelle proprietà in base alle quali nei ragionamenti, date certe proposizioni, altre seguono necessariamente in virtù di quelle* - Aristotele, *Topici*¹⁶.

Il termine inferenza indica una qualsiasi sequenza finita di proposizioni delle quali l'ultima è ottenuta come conclusione delle precedenti: queste sono chiamate premesse, mentre l'ultima proposizione è chiamata appunto conclusione. La logica studia quei particolari tipi di argomenti che sono inferenze, e lo fa in due modi: da un lato cerca di capire e dimostrare quando, dato un certo insieme di proposizioni, esse costituiscono una inferenza, producendone quindi la forma logica; dall'altro lato, cerca di mostrare quali proposizioni, o insiemi di proposizioni, corrispondono a certe forme di inferenze che vengono postulate.

I processi in base a cui vengono costruite inferenze, e vengono ricavate, inferite, derivate, nuove proposizioni (informazioni, risorse ecc.) a partire da proposizioni (informazioni, risorse ecc.) date, sono di fondamentale importanza per l'organizzazione delle nostre conoscenze e per l'attivazione

¹⁶ Si veda Marletti, Moriconi, Mariani (2009), p. 13 e seg.

delle strategie che conducono alla soluzione di problemi. Questi processi in logica sono chiamati dimostrazioni.

3.2 DIMOSTRAZIONI VS. REFUTAZIONI

Le inferenze possono essere considerate da due punti di vista opposti, o volendo usare una metafora spaziale, secondo due direzioni opposte: o come *dimostrazioni* di una data proposizione (o di un insieme di proposizioni), o come *confutazioni* di una data proposizione (o di un insieme di proposizioni). La prima prospettiva può essere rappresentata come un processo *top-down*, che si sviluppa dalle premesse alla conclusione; la seconda come un processo *bottom-up*, che risale dalla conclusione alle premesse.

Come viene evidenziato in Abrusci (2009)¹⁷ le dimostrazioni servono a *stabilire* proposizioni, mentre le refutazioni servono a *respingere* proposizioni. Una dimostrazione che stabilisce una proposizione A è detta dimostrazione di A. Una refutazione che respinge una proposizione A è detta refutazione di A. In genere, ci possono essere più dimostrazioni della stessa proposizione (fra le quali alcune sono più semplici, altre più complesse), e potrebbe non esserci alcuna dimostrazione di una data proposizione; analogamente, possono esserci più refutazioni della stessa proposizione, e potrebbe anche non esserci alcuna refutazione di una data proposizione. A partire da questi concetti di base, possiamo introdurre le seguenti definizioni formali, dove il simbolo ‘ \vdash ’ designa la relazione di dimostrabilità, a sinistra del quale si pongono le premesse della dimostrazione, e a destra la conclusione:

1. x è una dimostrazione di una proposizione A: $x \vdash A$
2. x è una refutazione di una proposizione A: $x: A \vdash$
3. una proposizione A è dimostrabile (A ha una dimostrazione): $\vdash A$
4. una proposizione A è refutabile (A ha una refutazione): $A \vdash$

Consideriamo alcuni esempi. Una dimostrazione di una proposizione come “Il gatto miagola e Maria apre la porta”, che contiene il concetto logico di congiunzione (AND) consiste nel produrre due dimostrazioni, una per la proposizione “Il gatto miagola” e un'altra per la proposizione “Maria apre la porta”; lo stesso vale per la dimostrazione di qualunque proposizione che abbia la forma congiuntiva “A e B”. Al concetto di congiunzione è associata la

¹⁷ Logica. Lezioni di primo livello, pp. 5-30.

regola: una dimostrazione della proposizione “A e B” è costituita da una coppia di dimostrazioni, una per la proposizione A e un'altra per la proposizione B¹⁸.

Una refutazione di una proposizione come “Il gatto miagola oppure Maria apre la porta” in cui compare il concetto logico di disgiunzione (OR), consiste nel produrre due refutazioni, una per la proposizione “Il gatto miagola” e una per la proposizione “Maria apre la porta”; lo stesso vale per la refutazione di qualunque proposizione che sia della forma “A oppure B”. Al concetto di disgiunzione è associata la regola: una refutazione della proposizione “A o B” è data da una coppia di refutazioni, una per la proposizione A e un'altra per la proposizione B¹⁹.

In modo analogo, una confutazione della proposizione “Se il gatto miagola, allora Maria apre la porta”, in cui compare il concetto logico di *condizionale* (se ... allora), può essere ricondotta ad una confutazione della proposizione “Il gatto non miagola oppure Maria apre la porta”, sulla base della equivalenza logica: $(A \rightarrow B) \leftrightarrow (\sim A \vee B)$. Una refutazione di quest'ultima proposizione consisterà nel produrre due refutazioni, una per la proposizione “Il gatto non miagola” e una per la proposizione “Maria apre la porta”.

3.3 DIMOSTRAZIONE DA IPOTESI

Si tratta di un concetto molto importante per la logica, che riguarda il modo in cui certe conclusioni possono essere ricavate (derivate, inferite) da un insieme di premesse. In generale, una dimostrazione di una proposizione A dalla proposizione B è anche detta *derivazione* di A da B, o *ragionamento* che porta alla proposizione A a partire dalla proposizione B, o *argomentazione* a favore di A sulla base di B. In una dimostrazione di una conclusione A da un'ipotesi, o premessa B, ogni singolo passo compiuto per ricavare la conclusione A dall'ipotesi B, o per ricondurre la conclusione A all'ipotesi B, viene chiamato *inferenza*. Su questa base, possiamo dare la seguente definizione:

¹⁸ Questa regola corrisponde alla condizione di verità del connettivo AND (\wedge) in base a cui una congiunzione $A \wedge B$ è *vera* se e solo se sia A che B sono *vere*.

¹⁹ Questa regola corrisponde alla condizione di verità del connettivo OR (\vee) in base a cui una disgiunzione $A \vee B$ è *falsa* se e solo se sia A che B sono *false*.

Una dimostrazione di A dalle ipotesi A_1, \dots, A_n è una dimostrazione di A dalla congiunzione delle proposizioni A_1, \dots, A_n ossia da $(A_1 \wedge \dots \wedge A_n)$.

Si noti che una dimostrazione di A da B , quando consiste nel ricavare la conclusione A dall'ipotesi B , consiste anche nel ricondurre la conclusione A all'ipotesi B : basta guardare all'indietro il processo compiuto nel ricavare A da B , ed esso diventa un processo che riconduce A a B . Dunque, una dimostrazione di una proposizione A da una proposizione B è sempre sia il *ricavare* A da B , sia il *ricondurre* A a B ²⁰.

Ci sono due modalità principali in base a cui considerare il processo associato all'atto di una dimostrazione. In un primo modo, una dimostrazione di una conclusione A da una ipotesi B (o insieme di ipotesi A_1, \dots, A_n) può essere considerata come l'atto di *ricavare* la conclusione A dall'ipotesi B (o dall'insieme di ipotesi A_1, \dots, A_n); questa modalità è detta dall'alto verso il basso, o "top-down" ed esprime il *movimento* di pensiero che va dalle ipotesi alla conclusione. Nell'altro modo, una dimostrazione di A da B è considerata come la *riconduzione* della conclusione A all'ipotesi B (o all'insieme di ipotesi A_1, \dots, A_n); questa modalità è detta dal basso verso l'alto o "bottom-up" ed indica il processo che risale dalla conclusione verso le premesse.

La presentazione dall'alto verso il basso (*top down*) di una dimostrazione di A da B corrisponde alla situazione in cui la dimostrazione è stata trovata e viene comunicata, esibita. D'altro lato, la presentazione dal basso verso l'alto di una dimostrazione di A da B (*bottom-up*) corrisponde al processo con il quale quella dimostrazione viene cercata e individuata: si parte dal problema A e lo si riconduce al problema B ²¹. Rappresentiamo così queste nozioni in modo formale:

- | | | |
|----|--|--------------------------------|
| 1. | x è una dimostrazione di A dall'ipotesi B | $x : B \mid - A$ |
| 2. | x è una dimostrazione di A dalle ipotesi A_1, \dots, A_n | $x : A_1, \dots, A_n \mid - A$ |
| 3. | A è dimostrabile dall'ipotesi B | $B \mid - A$ |
| 4. | A è dimostrabile dall'ipotesi A_1, \dots, A_n | $A_1, \dots, A_n \mid - A$ |

Consideriamo alcuni esempi. Nella psicologia del ragionamento sono studiati due importanti argomenti condizionali, conosciuti come Modus Ponens (MP)

²⁰ Abrusci (2009, p. 9)

²¹ Per approfondimenti si rimanda ad Abrusci (2009). Esempi di vari tipi di dimostrazioni e dimostrazioni da ipotesi sono presentati in Casadio (2005, Cap. II).

e Modus Tollens (MT), assieme alle loro fallacie, Affermazione del Conseguente (AC) e Negazione dell'Antecedente (NA).

MODUS PONENS

- | | |
|--|----------------------|
| 1. Se premo l'interruttore, la stanza si illumina. | 1. $A \rightarrow B$ |
| 2. Premo l'interruttore. | 2. A |
| \therefore La stanza si illumina. | $\therefore B$ |

MODUS TOLLENS

- | | |
|--|----------------------|
| 1. Se premo l'interruttore, la stanza si illumina. | 1. $A \rightarrow B$ |
| 2. La stanza non si illumina. | 2. $\sim B$ |
| \therefore Non premo l'interruttore. | $\therefore \sim A$ |

AFFERMAZIONE CONSEGUENTE

- | | |
|--|----------------------|
| 1. Se premo l'interruttore, la stanza si illumina. | 1. $A \rightarrow B$ |
| 2. La stanza si illumina. | 2. B |
| \neq Premo l'interruttore. | $\neq A$ |

NEGAZIONE ANTECEDENTE

- | | |
|--|----------------------|
| 1. Se premo l'interruttore, la stanza si illumina. | 1. $A \rightarrow B$ |
| 2. Non premo l'interruttore. | 2. $\sim A$ |
| \neq la stanza non si illumina. | $\neq \sim B$ |

Una dimostrazione dell'esempio di Modus Ponens qui considerato è l'argomento in base a cui la congiunzione delle proposizioni (le premesse dell'argomento) *Se premo l'interruttore, la stanza si illumina* e *Premo l'interruttore*, è condizione necessaria e sufficiente per ricavare la conclusione *La stanza si illumina*. D'altro lato, la proposizione (conclusione) *La stanza si illumina* può essere ricondotta in modo corretto alla premessa *Premo l'interruttore* e alla premessa *Se premo l'interruttore, la stanza si illumina*. Un ragionamento analogo può essere fatto con l'argomento Modus Tollens²².

Nel caso delle fallacie avremo invece una refutazione della loro conclusione a partire dalle premesse e diremo che le premesse non costituiscono una condizione necessaria e sufficiente per ricavare, ottenere, la conclusione; o, alternativamente che dalla conclusione non è possibile risalire in modo corretto alle premesse²³. Formalmente:

²² Una procedura automatica per dimostrare la validità di queste inferenze è costituito dal metodo delle tavole di verità: si prova che l'enunciato $(A_1 \wedge A_2) \rightarrow B$ ottenuto unendo la congiunzione delle premesse con la conclusione mediante il connettivo condizionale è una tautologia; si veda Casadio (2005, Cap. II).

²³ Con le tavole di verità si proverà che con entrambe le fallacie **non** si ottiene una tautologia.

Dimostrazione: $x: A \rightarrow B, A \mid - B$ Refutazione: $x: A \rightarrow B, \sim A \mid -$
 $x: A \rightarrow B, \sim B \mid - \sim A$ $x: A \rightarrow B, B \mid -$

3.3 DUALITÀ E INFORMAZIONE

La dualità pervade profondamente tutti gli ambiti della conoscenza e in particolare della logica ed è alla base della comunicazione tra le dimostrazioni e della loro dinamica. La dualità richiede la presenza di due punti di vista alternativi fra loro, con i quali si esprimono gli oggetti della nostra conoscenza o i contenuti dell'informazione. Si considerino i seguenti esempi.

ESEMPI DI CONCETTI DUALI

1. *venditore* vs. *acquirente* : uno stesso evento commerciale considerato da due punti di vista opposti, quello di chi *da*, ottenendo una certa cosa (p.es. una somma di danaro) e quello di chi *riceve*, producendo una certa cosa (p.es. una somma di danaro).
2. *output* vs. *input* : uno stesso dato può essere considerato come il risultato di un processo che lo *genera*, o come punto di partenza di un processo che lo *usa*.
3. I valori di verità “Vero” (1) e “Falso” (0) : sono impiegati in logica per esprimere punti di vista alternativi, e quindi la dualità; posso esprimere una informazione con una proposizione “vera” (se assumo il punto di vista di “dire la verità”), o con una proposizione “falsa” (se assumo il punto di vista di “dire la falsità”).
4. La coppia *dimostrazione* vs. *refutazione* è una coppia di punti di vista alternativi: una stessa informazione può essere considerata dimostrazione di una proposizione, e refutazione di un'altra proposizione, e viceversa.

Due proposizioni **A** e **B** sono dette logicamente duali quando sono duali rispetto alla coppia di punti di vista dimostrazione vs. refutazione: ogni dimostrazione dell'una è una refutazione dell'altra, e viceversa. Due proposizioni logicamente duali esprimono la stessa informazione sotto due punti di vista alternativi, quello della dimostrazione e quello della refutazione. $\sim A$ viene detto il duale di **A** (e viceversa).

Dimostrazione: $x \mid - 3 + 3 = 6$ Refutazione: $x: 3 + 3 \neq 6 \mid -$
 $x \mid - \sim(A \wedge \sim A)$ $x: A \wedge \sim A \mid -$

Formalmente: $x \mid - A$ è duale con $x: \sim A \mid -$

Dati due punti di vista (proposizioni) alternativi, cambiare una volta il punto di vista significa passare dall'uno all'altro punto di vista, e dunque esprimere un contenuto informativo con una proposizione che è duale rispetto a quella con la quale prima era stato espresso quello stesso contenuto. Cambiare due volte il punto di vista, equivale a ritornare al punto di vista iniziale: se abbiamo espresso qualcosa sotto un punto di vista mediante una proposizione A e poi cambiando il punto di vista abbiamo espresso lo stesso contenuto informativo mediante una proposizione B che è duale rispetto a A , cambiando di nuovo il punto di vista si ritorna alla proposizione A .

Questo duplice cambiamento di punto di vista nella logica classica è rappresentato dalla legge della Doppia Negazione in base a cui se neghiamo una proposizione A , otteniamo la sua duale $\sim A$, e negando di nuovo questa proposizione ritorniamo alla proposizione originaria A , formalmente:

$$\sim(\sim A) \rightarrow A \quad \text{e} \quad A \rightarrow \sim(\sim A) \quad \text{quindi} \quad A \leftrightarrow \sim(\sim A)$$

Nel linguaggio naturale la dualità viene di frequente introdotta attraverso concetti opposti come *sveglio / addormentato*, *vivo / morto*, *scapolo / sposato*, in cui ciascun elemento di ogni coppia è la negazione dell'altro. Questa presenza implicita della negazione svolge un ruolo importante nella cognizione umana, producendo interessanti effetti nel ragionamento e in certi casi errori. Ad esempio, per comprendere l'enunciato *se Maria è sveglia, allora non sogna*, bisogna essere consapevoli della regola implicita di significato: [dormire = non essere sveglio], che consente di inferire la conclusione *Maria non sogna* nell'argomento

1. Se Maria è sveglia, allora non sogna.
 2. Maria non dorme.
- \therefore Maria non sogna.

in base all'applicazione della regola della doppia negazione:

$$\begin{aligned} \text{non (dormire)} &= \text{non (non essere sveglio)} \\ \text{non (non essere sveglio)} &\leftrightarrow \text{essere sveglio.} \end{aligned}$$

ESERCIZI

E1. PROPOSIZIONI

- a. Fornire esempi di proposizioni semplici e proposizioni complesse.
- b. Fornire esempi di proposizioni con il connettivo *and*.
- c. Fornire esempi di proposizioni con il connettivo *or*.
- d. Indicare quali delle seguenti proposizioni sono complesse:
 1. Se piove non vado a visitare i miei parenti.
 2. Se piove Maria e Gianni restano a casa.
 3. Maria e Gianni restano a casa.
 4. Maria va a lezione oppure in biblioteca.
 5. Maria legge libri gialli o libri di fantascienza.

E2 ARGOMENTI

- a. Fornire esempi di argomenti relativi a situazioni quotidiane.
- b. Fornire esempi di argomenti relativi a problemi scientifici o psicologici.
- c. Fornire esempi di non-argomenti relativi a situazioni quotidiane o problemi psicologici.
- d. Dire quali dei seguenti sono argomenti:
 1. Le malattie dell'invecchiamento oggi possono essere diagnosticate diversi anni prima della loro manifestazione critica. Una diagnosi precoce delle malattie dell'invecchiamento (adulti di oltre 50 anni) aiuta a tenerle sotto controllo. Le persone che hanno superato i 50 anni possono evitare patologie croniche dell'invecchiamento.
 2. Le malattie dell'invecchiamento oggi possono essere diagnosticate precocemente. Le persone che hanno superato i 50 anni oggi godono di molti vantaggi nella vita quotidiana. Il numero delle persone anziane nella nostra società sta aumentando.
 3. Se i pirotti carulizzano elasticamente, allora carulizzano anche massicciamente. I pirotti carulizzano elasticamente. Quindi i pirotti carulizzano massicciamente.

E3 CONNETTIVI LOGICI E CIRCUITI

Abbiamo visto che un circuito elettrico che contiene solo interruttori a due posizioni (interruttore chiuso: passa la corrente (valore 1), interruttore aperto: la corrente non passa (valore 0) può essere rappresentato da una formula enunciativa del calcolo proposizionale.

ESERCIZIO: Scrivere i circuiti rappresentati dalle seguenti formule enunciative:

1. $(A \wedge B) \vee (C \wedge \neg A)$
2. $(A \wedge B) \vee ((C \vee A) \wedge \neg B)$
3. $(A \vee (C \wedge \neg B))$
4. $(\neg C \wedge A \vee ((C \vee \neg B) \wedge \neg C))$

E4 LEGGI DI DE MORGAN

Dimostrare le leggi di De Morgan usando le tavole di verità della logica proposizionale (si veda Casadio 2005, pp).

E5 CONNETTIVI LOGICI E OPERATORI BOOLEANI

L'operatore NOT restituisce il valore inverso del valore in entrata.

ESERCIZIO 1: descrivere il funzionamento della porta NOT e fornire esempi.

L'operatore OR restituisce il valore 1 se almeno uno degli elementi ha valore 1, mentre restituisce 0 in tutti gli altri casi.

ESERCIZIO 2: descrivere il funzionamento della porta OR e fornire esempi.

L'operatore AND dà come valore 1 se tutti gli operandi hanno valore 1, mentre restituisce il valore 0 in tutti gli altri casi.

ESERCIZIO 3: descrivere il funzionamento della porta AND e fornire esempi.

L'operatore XOR restituisce il valore 1 se e solo se uno solo dei due operandi è 1, e restituisce il valore 0 in tutti gli altri casi.

ESERCIZIO 4: descrivere il funzionamento della porta XOR e fornire esempi.

L'operatore NOR, la negazione del risultato dell'operazione OR, restituisce il valore 1 se e solo se tutti gli elementi sono 0, mentre restituisce 0 in tutti gli altri casi.

ESERCIZIO 5: descrivere il funzionamento della porta NOR e fornire esempi.

L'operatore XNOR è la negazione del risultato dell'operazione XOR; restituisce il valore 1 se tutti gli elementi hanno valore uguale a 1, oppure se tutti gli elementi hanno valore uguale a 0.

ESERCIZIO 6: descrivere il funzionamento della porta XNOR e fornire esempi.

ESERCIZIO 7: descrivere il funzionamento della porta seguente e indicare a quale connettivo corrisponde



ESERCIZIO 8: costruire le porte corrispondenti alle due leggi di De Morgan e mostrarne la relazione di dualità

ESERCIZIO 9: costruire una tabella in cui, nella prima parte, si vede come l'applicazione dell'operatore NOR ad A e B corrisponde alla applicazione dell'operatore AND alle negazioni $\sim A$ e $\sim B$; nella seconda parte, si vede come l'applicazione dell'operatore NAND a $\sim A$ e $\sim B$ corrisponde alla applicazione dell'operatore OR ad A e B.

E6 PORTE LOGICHE E RAPPRESENTAZIONE DELL'INFORMAZIONE

ESERCIZIO 1: Analizzando l'esempio relativo alla possibilità di fumare (§ 2.4), facendo riferimento al significato delle operazioni NOT, OR, AND, dire se la risposta alla domanda "Posso fumare" è SI o NO.

ESERCIZIO 2: Provare a costruire altre situazioni simili a quella dell'esempio, in cui vengono impiegate le operazioni logiche NOT, OR, AND, NOR, XOR, XNOR.

E7 INFERENZE E DIMOSTRAZIONI

ESERCIZIO 1: Richiamandosi al manuale di logica, fornire esempi di inferenze considerate dal punto di vista di una dimostrazione e da quello di una refutazione.

SUGGERIMENTO: costruire una dimostrazione di leggi logiche come il *Principio del Terzo Escluso*: $(p \vee \sim p)$ e una refutazione della loro negazione, ad esempio: $\sim (p \vee \sim p)$ (applicare le leggi di De Morgan)

ESERCIZIO 2: Fornire esempi di dimostrazioni da ipotesi.

SUGGERIMENTO: fare riferimento agli esempi ed esercizi del Cap. II in Casadio (2005), *Logica e Psicologia del Pensiero*.

ESERCIZIO 3: Fornire esempi di concetti duali. Fornire esempi di proposizioni duali.

CAPITOLO SECONDO

ONTOLOGIE

1. NOZIONI PRELIMINARI

L'Ontologia è quell'area della filosofia che si occupa dello studio dell'essere (*ontos*) dal punto di vista delle sue proprietà generali e indipendenti dalle sue determinazioni particolari²⁴. In una prospettiva cognitiva, una ontologia è la specificazione di una concettualizzazione condivisa degli elementi di un dominio, un modo per organizzare e categorizzare il mondo e le nostre conoscenze. Per tale ragione, gli studi sulle ontologie si collegano direttamente alla teoria degli insiemi²⁵, e sono alla base degli sviluppi delle ricerche informatiche e di intelligenza artificiale.

Tecnicamente una ontologia descrive il modo in cui diversi schemi relazionali (rappresentati da diagrammi, grafici, ecc.) vengono combinati in una struttura di dati (*data base*) contenente tutte le entità rilevanti e la specificazione delle loro relazioni in un certo dominio. I programmi informatici possono usare le ontologie per molti scopi come: classificazione e organizzazione delle informazioni, ragionamento induttivo e probabilistico, svariate modalità di risoluzione di problemi.

A partire dagli anni '50 del secolo scorso le ontologie sono state impiegate in vari ambiti di organizzazione della conoscenza, dall'informatica teorica agli studi nell'area della *Computer Science*: rappresentazione delle conoscenze, reti semantiche, *frames*, ecc.; abbiamo poi una grande varietà di applicazione nel campo dell'Intelligenza Artificiale (IA), dello sviluppo di sistemi esperti e della cibernetica.

La rappresentazione delle conoscenze prevede due momenti caratteristici: una fase descrittiva, tassonomica, che consiste nella descrizione delle conoscenze, la loro selezione, definizione e formalizzazione. E una fase operativa, applicativa, che consiste nell'uso delle informazioni rappresentate, in base agli scopi specifici, particolari o generali, che si vogliono raggiungere.

²⁴ In merito alle concezioni relative all'essere e alle sue determinazioni si richiamano in primo luogo i pensatori della filosofia greca come Eraclito, Parmenide, Platone, Aristotele; si veda Reale e Antiseri (2009).

²⁵ Per le nozioni elementari di *Teoria degli Insiemi* si veda Casadio (2005), *Logica e Psicologia del Pensiero* (Cap. I, pp. 35-37), in particolare: insieme, classe, proprietà, relazione, funzione, unione e intersezione; si vedano anche Abrusci (2009), Mangione e Bozzi (1993), Mendelson (1972).

2. CATEGORIZZAZIONE E PERCEZIONE

L'organizzazione delle conoscenze e lo studio delle proprietà dei concetti si basano sulla attività teorica e cognitiva nota in letteratura come *categorizzazione*. Definiamo categoria un insieme di oggetti (nel senso ampio del termine) individuati da una o più *proprietà* distintive.

ESEMPIO: la categoria TAVOLO è individuata da tutti gli oggetti che: hanno una superficie piana, abbastanza ampia, su cui ci si può appoggiare, sostenuta da almeno tre, in genere quattro, elementi, chiamati "gambe". A questa categoria appartengono i tavoli grandi, i tavoli nella norma ed i tavoli piccoli o "tavolini"; ma non vi apparterranno altri mobili con sostegni e superfici piane come comodini, scrivanie.

Le nostre conoscenze (competenza cognitiva) sono organizzate in modo tale da riflettere le relazioni tra i concetti attraverso una rete articolata di relazioni, che la filosofia, la psicologia e le scienze cognitive hanno rappresentato in varie maniere, dando luogo a diversi modelli teorici della categorizzazione. Tali relazioni possono essere in particolare:

- *Semantiche* : cognitive, relative al contenuto, culturali;
- *Tassonomiche* : ottenute su base classificatoria
p. es.: mammifero → felino → gatto;
- *Associative* : organizzazione di conoscenze per associazione
p. es.: gatto & colore nero → gatto nero.

2.1 ALBERO DI PORFIRIO

Per analizzare le conoscenze, le loro relazioni e articolazioni, sono utili le rappresentazioni mediante diagrammi e grafi; una di queste è l'*Albero di Porfirio*²⁶, un grafo, costruito come le biforcazioni di un albero, al fine di rappresentare la coordinazione e subordinazione delle categorie coinvolte, ad esempio dei generi e delle specie. Come mostra la Fig. 3, muovendo dal genere sommo si scende fino alle specie più basse mediante una suddivisione per dicotomia: la SOSTANZA si divide in CORPOREA e INCORPOREA, quella

²⁶ Il nome deriva dal suo impiego nel secondo capitolo dell'*Isagoge* di Porfirio (Tiro, 233/234 - Roma, 305; filosofo e teologo greco). Questo tipo di grafo ad albero si presta bene alla rappresentazione dialettica dei concetti, come nel noto esempio del *pescatore con la lenza* che compare nel dialogo il *Sofista* di Platone.

corporea in ANIMATA e INANIMATA, quella animata in SENSIBILE e INSENSIBILE, ecc.:

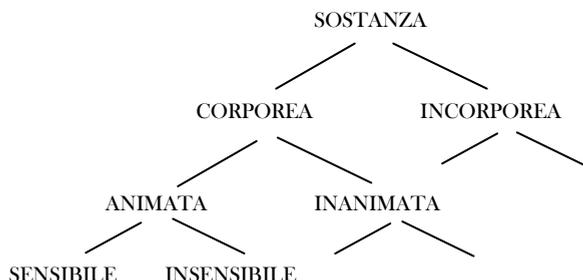


FIGURA 3. *Albero di Porfirio di generi e specie*

La filosofia, la psicologia e le scienze cognitive in genere, hanno elaborato una varietà di modelli teorici della categorizzazione, tra i quali in particolare ricordiamo: il modello della teoria classica, la teoria del prototipo e il modello funzionalista.

2.2 TEORIA CLASSICA

Secondo la teoria classica un concetto è specificato da un insieme di proprietà, dette anche tratti o *features*, che rappresentano le condizioni necessarie e sufficienti per determinare l'appartenenza alla categoria ad esso associata²⁷. Ad esempio, perché una certa persona sia classificata come “nonna” deve possedere proprietà come *essere di sesso femminile* e *essere genitore di un genitore* (condizione necessaria); tali proprietà sono condizioni sufficienti perché l'individuo in questione venga assegnato alla categoria NONNA²⁸:

NONNA

²⁷ Questo tipo di analisi rimanda alla distinzione delineata da G. Frege e riproposta successivamente da R. Carnap, tra *sensu* o *intensione* di un concetto e la sua *denotazione* o *estensione*: la prima dimensione è rappresentata dall'insieme di proprietà distintive che caratterizzano il concetto, mentre la seconda consiste nell'insieme degli oggetti o individui che soddisfano tali proprietà. Nei termini di tale dicotomia, il significato è determinato come nesso tra *sensu* (intensione) e *denotazione* (estensione); si vedano Frege (1892) e Carnap (1947).

²⁸ Per la teoria classica della categorizzazione si veda Bruner, Goodnow, Austin e Brown (1956); si vedano anche Rosch (1983), Macchi (1999).

individuo di sesso femminile
 essere genitore di un genitore

formalmente: $\forall(x)[N(x) \leftrightarrow FG(G(x))]$

dove N è un predicato che designa il concetto o categoria NONNA, F designa la proprietà FEMMINILE, G designa la proprietà GENITORE, e la proprietà di secondo ordine $G(G(x))$ designa GENITORE DI UN GENITORE. Possiamo leggere questa espressione come l'asserzione: "per ogni individuo x, nonna di x equivale a genitore di genitore di x, di sesso femminile". Su questa base, un concetto, o categoria, può essere rappresentato da un elenco congiuntivo di attributi, come quelli mostrati nell'esempio.

Poiché gli attributi o specificazioni (*features*) rappresentano proprietà che si applicano ad elementi che appartengono ad una determinata categoria, i confini tra una categoria e l'altra sono ben delineati e definiti. L'appartenenza di un elemento ad una certa categoria è precisa, controllata.

Questa concezione ha ricevuto ampio consenso fino agli anni 1970-80, poiché si tratta di una teoria economica, che consente definizioni precise; tuttavia essa ha il limite di non riuscire a rendere conto di molti concetti naturali, delle categorie sfumate e delle relazioni tra categorie. Questo aspetto è stato bene evidenziato da L. Wittgenstein quando parla di *somiglianze di famiglia*: gli oggetti, o i fenomeni, che possono essere raccolti sotto una certa categoria non sono separabili in base a confini netti, ma sono *imparentati* gli uni con gli altri in molti modi diversi. "Considera, ad esempio, i processi che chiamiamo "giochi" [...] giochi da scacchiera, giochi di carte, giochi di palla, [...] Che cos'è in comune a tutti questi giochi? [...] se li osservi, non vedrai certamente qualche cosa che sia comune a *tutti*, ma vedrai delle somiglianze, parentele, e anzi ne vedrai tutta una serie"²⁹.

Prendendo dunque in esame la collezione di oggetti, o individui, che designiamo come "membri di una categoria", osserviamo che nella maggioranza dei casi essi sono correlati da insiemi di condizioni, o proprietà, che si sovrappongono, quelle che Wittgenstein chiama *somiglianze di famiglia*, piuttosto che essere identificati da precise condizioni, o proprietà, comuni.

Possiamo così sintetizzare le principali obiezioni rivolte alla teoria classica:

- i. Criterio di *identificazione* delle proprietà o tratti che caratterizzano una categoria e la distinguono da altre categorie simili. Ad esempio, la categoria TIGRE è caratterizzata dal possesso di proprietà generiche come ANIMALE, MAMMIFERO, FELINO, ma anche da proprietà specifiche come avere il manto di colore *giallastro*

²⁹ Wittgenstein (1953), trad. it. pp. 46-47.

e *striato di scuro*. Tuttavia, esistono tigri albine, il cui pelo non presenta la caratteristica striatura e colorazione. Questi esemplari, come in generale le eccezioni al soddisfacimento di una particolare proprietà specifica come il colore del manto, conservano tuttavia le proprietà generali (i caratteri genetici) della propria specie, per cui si tratta sempre di *tigri*. La teoria classica non è in grado di fornire una spiegazione di queste variazioni.

- ii. Criterio di *appartenenza* ad una categoria. Sono frequenti i casi in cui non si trova un accordo sulla classificazione di un oggetto, vi sono casi incerti o non chiari come *tappeto*, che può essere assegnato alla categoria MOBILI o a quella dei TESSUTI, oppure *taccuino*, che può essere assegnato alla categoria QUADERNI, oppure a quella dei LIBRI.
- iii. Criterio della *tipicità*. I membri di una stessa categoria non condividono tutti le stesse proprietà caratteristiche, ma si incontrano esemplari più tipici rispetto ad altri. Ad esempio un *canarino* sarà considerato un esemplare più tipico della categoria UCCELLO di un *pinguino* o di uno *struzzo*. Nella attribuzione di appartenenza ad una categoria non vale dunque un criterio di omogeneità, per cui tutti gli elementi sono alla pari come dovrebbe essere previsto dalla definizione della teoria classica di identità categoriale come cadere sotto uno stesso concetto.
- iv. Criterio della *ereditarietà*. Secondo la teoria classica un concetto possiede tutte le proprietà della categoria da cui dipende (categoria *sovraordinata*); ad esempio, *animale* → *mammifero* → *uomo*. Vi sono tuttavia esemplari che presentano un minor numero di proprietà distintive in comune con la categoria *sovraordinata* rispetto a categorie più lontane; di nuovo un *pinguino* avrà meno proprietà in comune con la categoria UCCELLO, da cui dipende, rispetto a categorie più generali e lontane come ANIMALE.

2.3 TEORIA DEL PROTOTIPO

Un modello alternativo alla teoria classica è rappresentato dalla Teoria del Prototipo (*Prototype Theory*) elaborata alla fine degli anni '70 del secolo scorso dalla psicologa statunitense Eleanor Rosch³⁰. Secondo tale teoria, le categorie naturali sono organizzate attorno ad un esemplare migliore, il *prototipo*, che costituisce il rappresentante medio, lo standard, rispetto al quale giudicare l'appartenenza o meno alla categoria in questione. In base a questa ipotesi, le proprietà rappresentative non devono necessariamente valere per tutti gli elementi assegnati ad una certa categoria, ma solo per il prototipo. Una categoria è organizzata secondo una struttura gerarchica, che si sviluppa su più dimensioni o livelli, rispetto ai quali il prototipo occupa la

³⁰ Si vedano Rosch (1977), (1978), (1983).

zona centrale o *core*. Ad esempio, la categoria FELINO può essere rappresentata con il seguente schema:

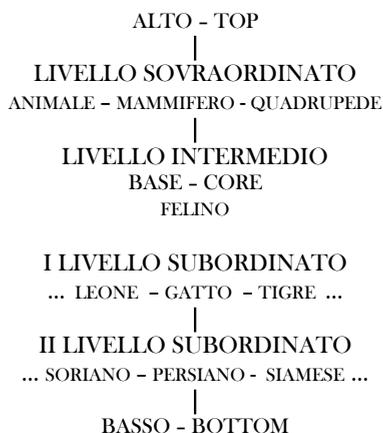


FIGURA 4. *Prototipo della categoria: FELINO*

Al livello *sovrordinato*, nella parte alta della gerarchia, troviamo le categorie generali come ANIMALE, MAMMIFERO, ecc., i cui membri hanno in comune un numero limitato di proprietà caratteristiche, ad esempio *vivente*, *animato*; la variabilità interna di queste categorie è molto ampia, poiché vi sono animali che non sono mammiferi, o mammiferi che non sono quadrupedi.

Al livello intermedio, o *base*, troviamo i concetti che hanno molte proprietà in comune tra loro e poche in comune con altre categorie dello stesso livello. Ad esempio, FELINO si differenzierà in modo radicale da CANIDE pur condividendo il livello sovraordinato MAMMIFERO. I concetti *base* sono linguisticamente i più significativi della tassonomia, vengono appresi per primi e sono elaborati più velocemente, come ha mostrato la ricerca sperimentale³¹. Dei concetti *base* è possibile rappresentarsi un esemplare medio, il prototipo appunto, come il concetto di *felino* tipico.

Al livello *subordinato*, infine, vengono specificati diversi insiemi di proprietà caratteristiche della categoria base, ad esempio quelle che caratterizzano vari tipi di felini come *gatto*, *leone*, ecc., oppure, restringendo l'attenzione alla categoria *gatto*, vari tipi di gatti come *soriano*, *siamese*, ecc.; a questo livello si riduce la variabilità tra categorie: i membri di ogni categoria subordinata condividono molti attributi con quelli delle altre categorie subordinate dello

³¹ Si vedano i risultati sperimentali presentati nel capitolo *Formazione dei concetti e processi di categorizzazione* di L. Macchi, in Girotto e Legrenzi (1999, pp. 91-115).

stesso livello; si presenta spesso un alto grado di somiglianza come quello tra un gatto e una lince, o tra i gatti delle varie famiglie.

L'articolazione gerarchica di una famiglia di categorie che si organizza attorno ad un prototipo, ed i diversi gradi di correlazione degli attributi ammessi dalle categorie generali al livello sovraordinato, rispetto a quelle al livello base o subordinato, consentono di risolvere molti dei problemi della teoria classica, come la *tipicità* o l'*ereditarietà*.

Anche la teoria del prototipo ha incontrato, tuttavia, diversi problemi, collegati in particolare al criterio della *somiglianza*, che non sempre si dimostra adeguato per la definizione di esemplare migliore o esemplare tipico che possiede il numero maggiore di attributi distintivi di una categoria. In molti casi il prototipo è individuato da ragioni di tipo funzionale, legate all'uso, alla pratica sociale o culturale. Ad esempio, per la categoria VESTIARIO, in cui attributi distintivi sono *coprire, tenere caldo*, genere *femminile vs. maschile*, il prototipo sarà l'esemplare che meglio svolge tali funzioni, e che viene usato più frequentemente a tale scopo, come *pantaloni*, nel caso maschile, o *gonna*, nel caso femminile³².

2.4 TEORIA FUNZIONALISTA

Lo psicologo statunitense Lawrence W. Barsalou³³ ha sollevato varie obiezioni alla teoria strutturalista del prototipo, proponendo un modello funzionalista che postula che la categorizzazione non è un processo stabile, ma variabile e dipendente da contesto: persone diverse, ma anche la stessa persona, possono formarsi concetti diversi relativamente ad una stessa categoria di oggetti in relazione alla loro occorrenza in contesti e situazioni diverse. Per esempio, se si considera la categoria MOBILI nel contesto di una cucina, *fornello* sarà più tipico di *scrivania*, mentre se si considera la stessa categoria nel contesto di un ufficio, *scrivania* diventa il concetto più tipico.

In questa prospettiva, che valorizza l'uso, lo scopo e altre motivazioni funzionali delle categorie, i concetti non sono più stabili, ma dipendono in modo rilevante dal contesto in cui vengono elaborati ed impiegati. Per la categorizzazione si dimostra fondamentale l'organizzazione delle conoscenze, così come sono rappresentate nella mente dai concetti. Una chiara analisi di questa situazione è fornita da Marvin Minsky:

³² Si vedano Rosch (1983), Rosch e Mervis (1975), Barsalou (1987).

³³ Vedi Barsalou (1987).

“La vecchia idea di classificare le cose in base alle proprietà non è del tutto soddisfacente, poiché tanti differenti tipi di qualità interagiscono tra loro secondo modalità complicate. Ogni situazione o condizione di cui abbiamo esperienza è influenzata o, se vogliamo, colorata, da migliaia di sfumature e variazioni contestuali, nello stesso modo in cui guardare attraverso un vetro colorato ha effetti su ciò che vediamo. [...] Alcune di queste condizioni e relazioni possono corrispondere a parole del linguaggio, ma nella maggioranza dei casi non abbiamo parole per esprimerle, così come non abbiamo forme linguistiche per esprimere in modo esatto una grande varietà di odori e sapori, gesti e intonazioni, atteggiamenti e disposizioni”³⁴.

Minsky indica la seguente tipologia delle possibili proprietà contestuali che influenzano la comprensione e classificazione dei concetti³⁵:

- *materiali* : animato, inanimato, naturale, artificiale, ideale, attuale
- *perceptive* : colore, consistenza, sapore, sonorità, temperatura
- *solidità* : durezza, densità, flessibilità, robustezza
- *forma* : angolosità, curvatura, simmetria, verticalità
- *permanenza* : rarità, durata, fragilità, sostituibilità
- *locazione* : ufficio, casa, veicolo, teatro, città, foresta, fattoria, ...
- *posizione* : all'interno, all'esterno, pubblica, privata, ...
- *attività* : cacciare, giocare, lavorare, divertirsi, ...
- *relazione* : cooperazione, conflitto, negoziazione, confronto, ...
- *sicurezza* : salvezza, pericolo, rifugio, fuga, sconfitta, ...

Così in un contesto in cui è saliente la collocazione storica, la parola *Boston* porterà alla mente immagini e concetti relativi alla Rivoluzione Americana, mentre in un contesto differente, in cui sono rilevanti informazioni di tipo geografico, la parola evocherà una città degli Stati Uniti d'America; un ulteriore sfondo potrà evocare invece una famosa università, e così via.

In particolare, i concetti teorici, formulati per ragioni classificatorie, non saranno mai identici ai concetti reali, elaborati dagli individui nelle varie situazioni in cui si trovano. Mentre i primi sono essenzialmente indipendenti dal contesto, sono delle idealizzazioni che vengono poste alla base della organizzazione delle nostre conoscenze, i secondi dipendono in modo cruciale dalle variazioni del contesto e dell'ambiente.

³⁴ *The Society of Mind* (1985, p. 211). Marvin Lee Minsky (New York 1927), informatico e scienziato cognitivo statunitense, ha sviluppato ricerche nell'ambito dell'intelligenza artificiale presso il MIT, in particolare nel campo delle reti neurali e delle strutture di organizzazione della conoscenza come i *frames*.

³⁵ *Ibid.* Queste modalità contestuali che intervengono nella formazione dei concetti richiamano le categorie della ontologia aristotelica e le categorie kantiane, come *qualità*, *quantità*, *modo*, *relazione*; si vedano in proposito anche il *Principio di Cooperazione* e le regole conversazionali in Grice (1975).

3. RAPPRESENTATIVITÀ

Poiché gli esseri umani basano i loro giudizi, relativi alle occorrenze di un evento, sul grado di corrispondenza tra le caratteristiche di tale evento e le proprietà della categoria cui appartiene, gli esempi rappresentativi tendono ad essere giudicati più frequenti degli esempi non rappresentativi³⁶.

Negli studi sulla categorizzazione (e.g. Rosch 1978) si è dimostrato che gli oggetti *più rappresentativi* di una categoria sono quelli che più velocemente sono giudicati appartenere a tale categoria: un *canarino* è più rappresentativo della categoria UCCELLI di uno *struzzo*.

Giungiamo in tal modo a parlare di *rappresentazioni*. Una rappresentazione è qualcosa che sta per qualcos'altro. Abbiamo due domini o mondi messi in relazione: il mondo *rappresentato* vs. il mondo *rappresentante*. Il sistema rappresentazionale del secondo fornisce gli elementi in base ai quali analizzare, ricostruire, conoscere, il primo. Le rappresentazioni servono quindi a capire, disegnare, le proprietà del mondo rappresentato. I sistemi di rappresentazione forniscono informazioni relative al mondo che costituisce il loro oggetto, secondo due modalità:

- Stesse informazioni : modi diversi di organizzarle
- Informazioni diverse : stesso sistema per coglierle, esprimerle

Nel primo caso abbiamo una informazione considerata secondo più sistemi rappresentazionali, nel secondo, abbiamo più informazioni considerate secondo uno stesso sistema rappresentazionale. Ad esempio, il modello *idraulico* ha costituito una rappresentazione appropriata del sistema di circolazione sanguigna umano, per cui si parla del cuore come una *pompa*, delle vene come *condotti*, ecc. Tuttavia, l'organizzazione della circolazione sanguigna potrebbe essere rappresentata anche mediante un modello più astratto, ad esempio un *reticolo* di relazioni.

D'altro lato, il modello idraulico può essere usato per rappresentare anche altri oggetti, ad esempio il sistema di comunicazioni all'interno di una catena commerciale, per cui si parla di *flusso* di clienti, *canale* di vendita, ecc.³⁷.

³⁶ Si vedano in proposito gli studi di Kahneman e Tversky (1996), Kahneman (2000), Kahneman e Frederick (2002).

³⁷ Nella costruzione di un sistema rappresentazionale esistono tre tipi fondamentali di modelli: (i) *in scala*, (ii) *analogici*, (iii) *teorici*; i primi forniscono una rappresentazione fedele del mondo rappresentato, ma con una diversa modalità di misura (scala maggiore vs. minore rispetto all'originale), ad esempio il modellino di una nave; i secondi mantengono l'analogia con il loro oggetto, ma rimandano a sistemi materiali diversi, come nel caso del sistema idraulico; infine, i terzi instaurano una relazione puramente formale con l'oggetto che deve essere rappresentato.

3.1 MODELLI RAPPRESENTAZIONALI

Nel seguente esempio³⁸ è mostrato in (A) come una stessa informazione, il peso umano (mondo rappresentato), possa essere l'input di due diversi sistemi rappresentazionali, numerico e mediante segmenti; in (B), invece, è mostrato come due diversi tipi di informazione, il peso umano e l'altezza di certi rettangoli, possano essere rappresentati mediante il solo sistema rappresentazionale dei segmenti.

A) INFORMAZIONE	SISTEMA RAPPRESENTAZIONALE
<i>peso umano</i>	i) in base a numeri ii) in base a segmenti
B) INFORMAZIONE	SISTEMA RAPPRESENTAZIONALE
i) peso umano ii) altezza rettangoli	<i>in base a segmenti</i>

Assumendo di considerare un gruppo di tre individui a cui assegniamo i pesi corporei sotto riportati e associamo ad essi le relative misure in segmenti, il caso (A) diventa:

Informazione	→	Rapp. (i)	∧	Rapp. (ii)
individuo: a		peso: 20		segmento: ____
individuo: b		peso: 30		segmento: _____
individuo: c		peso: 50		segmento: _____

mentre il caso (B) diventa:

Informazione (i)	∧	Informazione (ii)	→	Rappresentazione
peso: 20		altezza: 20		segmento: ____
peso: 30		altezza: 30		segmento: _____
peso: 50		altezza: 50		segmento: _____

I modelli di rappresentazione possono essere confrontati in base a diversi criteri come la modalità di definizione, la scelta tra modalità “top-down” vs. “bottom-up”, le dimensioni di “staticità” vs “dinamicità”. In particolare avremo:

- Modelli dichiarativi: reti semantiche, tassonomie, ontologie, ...
- Modelli descrittivi: Frames, Scripts, ...
- Modelli statistico-probabilistici: reti neurali, ...

³⁸ Si veda Giroto e Legrenzi (1999).

3.2 RETI SEMANTICHE

Tra i modelli di rappresentazione della conoscenza, quelli che hanno maggiore impiego in informatica e intelligenza artificiale sono le reti semantiche (*semantic nets* o *semantic networks*)³⁹.

Le reti semantiche sono grafi orientati formati da *vertici* (o nodi), che rappresentano concetti, e *archi* (*edges*), che rappresentano relazioni semantiche tra i concetti. Sono dizionari elettronici, che sfruttano le proprietà della teoria dei grafi, e possono essere letti da una macchina (elaboratore automatico). Le reti semantiche sono impiegate nell'ambito della deduzione logica automatizzata, nello studio dei grafi concettuali, e in vari modelli di rappresentazione delle conoscenze lessicali e traduzione automatica. Il seguente è un semplice esempio di rete con sei vertici e sette archi:

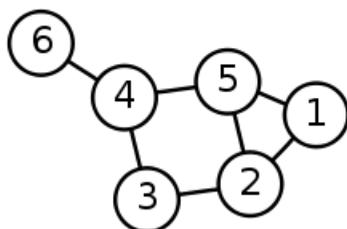


FIGURA 4. Grafo con 6 vertici e 7 archi

Una caratteristica importante delle reti semantiche è la loro capacità di rappresentare le relazioni di dominanza e di dipendenza:

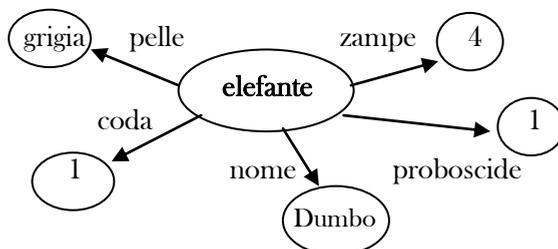


FIGURA 5. Rete semantica di ELEFANTE

³⁹ Le reti semantiche per i calcolatori sono state introdotte nel 1956 da R. H. Richens per le ricerche sulla traduzione automatica dei linguaggi naturali; sono state poi sviluppate negli anni '60, in particolare da M. R. Quillian, per vari impieghi in ambito computazionale e di intelligenza artificiale.

Un noto esempio di rete semantica è WordNet⁴⁰, un ampio data base lessicale della lingua inglese, dotato di semantica associativa. Nomi, verbi, aggettivi e avverbi sono raggruppati in insiemi di sinonimi cognitivi, ciascuno dei quali esprime un concetto distinto. I collegamenti avvengono tramite relazioni di tipo lessicale e semantico. Il sistema risultante di parole collegate tra loro in modo significativo può essere visitato su Internet.

È possibile rappresentare proprietà logiche per mezzo di reti semantiche, come i grafi concettuali, introdotti da John F. Sowa per rappresentare gli schemi concettuali usati nei data base, che hanno una capacità espressiva pari al calcolo dei Predicati del Primo Ordine⁴¹.

Per ogni modello di rappresentazione è stato sviluppato un tipo appropriato di ragionatore, un programma in grado di usare le conoscenze raccolte in una Base di Conoscenza (KB) per estrarne di “nuove”. I più fortunati abbinamenti di KB + ragionatore sono venduti come sistemi esperti.

I ragionatori attualmente utilizzati con le ontologie possono essere ricondotti direttamente ai primi *Theorem prover* dell'Intelligenza Artificiale (IA) che, programmati per applicare regole d'inferenza, ricevevano una serie di assiomi in input e restituivano teoremi. Rispetto a questi, alle regole logiche canoniche sono aggiunte spesso delle ottimizzazioni per migliorare le prestazioni di calcolo e per “aggiustare” le possibili risposte.

4. ORGANIZZAZIONE DELLE CONOSCENZE TRAMITE ONTOLOGIE

L'impiego fondamentale delle ontologie è nell'ambito dell'organizzazione delle conoscenze⁴². Da un punto di vista generale possiamo dire che una ontologia è utile quando:

- si intende guardare ad una stessa classe di risorse da punti di vista differenti (molteplicità di punti di vista);

⁴⁰ WordNet è stato realizzato presso la Princeton University e assomiglia a un Thesaurus, nel senso che raccoglie insieme le parole in base al loro significato; tuttavia, le parole sono raggruppate insieme non solo per la somiglianza di significato, ma anche in base ad altre relazioni semantiche e strutturali significative.

⁴¹ Vedi Sowa (1984). Tra i vari tipi di reti semantiche utilizzate per l'elaborazione di conoscenze lessicali, ricordiamo il *Semantic Network Processing System* (SNePS), di S. C. Shapiro, il paradigma MultiNet (*Multilayered ampliata Semantic Network*), per la rappresentazione della semantica del linguaggio naturale e le applicazioni di *Natural Language Processing*. Su quest'ultimo tema si vedano De Vincenzi (1991), De Vincenzi e Lombardo (2000).

⁴² Per approfondimenti si rimanda a Romano (2011).

- si prevede la necessità di cambiare alcune relazioni tra le risorse trattate e in generale di dover modificare la classificazione adottata (p. es. in seguito a nuove scoperte o per ottenere una migliore comprensione di certi fenomeni).

Un'ontologia è composta da due elementi: una **T-Box** (Scatola dei termini) e una **A-Box** (Scatola delle asserzioni) che insieme formano la “Knowledge Base” (KB). La **T-Box** (*box of Terminology*) è la parte tassonomica dell'ontologia, dove sono definiti i concetti, i termini usati per denominarli, e le proprietà e relazioni che possono sussistere tra loro. In breve, è la parte intensionale dell'ontologia. La **A-Box** (*box of Assertions*) è la parte dell'ontologia che contiene i fatti, dove cioè le singole istanze (oggetti, individui) sono classificate come appartenenti ad una precisa classe e dove sono indicate le proprietà definite per la classe di appartenenza di ciascuna istanza; in parole povere è la “sostanza” dell'ontologia.

Una ontologia consta di:

- una lista di termini (o concetti) gerarchicamente ordinati, i.e. una tassonomia;
- definizioni formali dei termini (espresse in un certo formalismo logico);
- proprietà e relazioni tra termini [la **T-Box**];
- una lista di assegnazioni di tipi a risorse e istanziazioni di proprietà e relazioni [la **A-Box**].

Le Ontologie assolvono ad alcune funzioni essenziali:

- condividere conoscenze comuni delle strutture di informazione (tra soggetti, tra agenti software);
- permettere il riuso di domini di conoscenza;
- fare assunzioni esplicite sul dominio (per facilitare la comprensione, l'aggiornamento, i cambiamenti).

Per organizzare le conoscenze tramite ontologie ci si basa su operazioni di *concettualizzazione*: rappresentazione formale della realtà come percepita e organizzata da un agente, indipendentemente dalla modalità di rappresentazione che viene impiegata, in particolare la lingua che viene usata (*Vocabolario* utilizzato) e dalle particolari situazioni. Situazioni differenti, che coinvolgono gli stessi oggetti, descritti da differenti vocabolari, devono condividere la stessa concettualizzazione.

Una ontologia può quindi essere definita come una specificazione esplicita, formale, di una concettualizzazione condivisa. Deve poter essere compresa da una macchina; vengono definiti in maniera esplicita sia i concetti che i vincoli

sul loro uso; una ontologia cattura conoscenze collettive (non private, ma accettate da gruppi di individui); è un modello astratto di descrizione dei concetti rilevanti di fenomeni del mondo reale.

Le ontologie svolgono un ruolo centrale nella realizzazione del progetto di organizzazione della conoscenza chiamato “Semantic Web”⁴³. Il Web Semantico rappresenta l’ambizioso progetto di arricchire la rete di Internet con un ulteriore livello semantico (i metadati) per permettere agli utenti di usare le informazioni e impiegarle per la realizzazione di vari scopi.

Il Web Semantico rappresenta l’evoluzione di Internet da semplice sistema di comunicazione e recupero di informazioni e documentazioni, ad un sistema “intelligente” in cui l’informazione è compresa da specifici software in grado di cooperare e assistere l’utente in compiti complessi. Si tratta di una nuova forma di conoscenza: flessibile, riutilizzabile, condividibile, dove *condivisione* significa che differenti applicazioni usano le stesse risorse, e *riutilizzazione*, significa che componenti già esistenti vengono impiegate per costruire nuove applicazioni.

Il ruolo del Web Semantico è stato efficacemente anticipato da uno dei suoi creatori, Tim Berners-Lee, nel 1994:

“Adding semantics to the Web involves two things: allowing documents which have information in machine-readable forms, and allowing links to be created with relationship values. Only when we have this extra level of semantics will we be able to use computer power to help us exploit the information to a greater extent than our own reading”⁴⁴.

⁴³ Il *World Wide Web Consortium* (W3C) ha iniziato la ricerca sul Web Semantico intorno al 2000. Il Web Semantico fa ricorso a vari linguaggi, a partire dall’originario HTML, come XML, RDF, OWL.

⁴⁴ La citazione è tratta da Romano (2011, p. 11).

ESERCIZI

E1 CATEGORIZZAZIONE

ESERCIZIO 1:

Fornire esempi, collegando proprietà della percezione con la categorizzazione e, più in generale, l'organizzazione della conoscenza. Ricavare conclusioni interessanti dal punto di vista della psicologia, delle scienze cognitive e delle neuroscienze.

ESERCIZIO 2:

Abbiamo visto come una informazione può essere considerata secondo più sistemi rappresentazionali, oppure come più informazioni possano essere considerate secondo uno stesso sistema rappresentazionale. Fornire esempi del primo e del secondo caso

ESERCIZIO 3:

Rappresentiamo il mondo dei rettangoli usando segmenti. La lunghezza dei segmenti deve rispettare le relazioni esistenti fra le dimensioni dei rettangoli: altezza vs. lunghezza. Avremo due rappresentazioni diverse in base al fatto che la dimensione preservata sia: ALTEZZA vs. LUNGHEZZA.

E2 SISTEMI RAPPRESENTAZIONALI

ESERCIZIO 1:

Ricerca informazioni sul concetto di FRAME e produrre esempi.

ESERCIZIO 2:

Ricerca informazioni sul concetto di SCRIPT e produrre esempi.

ESERCIZIO 3:

Elaborare esempi di reti semantiche che rappresentino le seguenti situazioni:

- (a) una sedia a sdraio di vimini;
- (b) Piero è amico di Maria e fratello di Carlo;
- (c) Giovanni dà il libro a Maria;
- (d) Bilbo trova l'anello magico nella caverna di Gollum.

E3 ONTOLOGIE

ESERCIZIO 1:

- a. Fornire esempi di ontologie tratte in particolare da ambienti della psicologia e delle scienze della vita.
- b. Fornire esempi di rappresentazioni strutturali di ontologie (grafi, diagrammi, ecc.).
- c. Fare ricerche su Internet per trovare esempi di data base e applicazioni del concetto di Web Semantico.

ESERCIZIO 2:

Elaborare esempi di ontologia relativi allo studio della genetica, in riferimento a (i) genoma e classificazione dei geni; (ii) correlazione con le espressioni fenotipiche; (iii) correlazione con la diagnostica clinica (espressioni patologiche)

PARTE SECONDA

INFORMAZIONE E AUTOMAZIONE

CAPITOLO TERZO

INFORMAZIONE

1. TRATTAMENTO SIMBOLICO DELLE INFORMAZIONI

In questa sezione prenderemo in esame il modo in cui le informazioni possono essere elaborate e gestite sistematicamente al fine di ottenere risultati certi e generali. Una nozione fondamentale in quest'ambito è quella di *algoritmo*. L'algoritmica non è semplicemente un ramo dell'informatica, ma ne rappresenta il nucleo e per sua natura si dimostra fondamentale per tutte le discipline che si avvantaggiano dell'impiego dei computer: non solo le discipline ingegneristiche e tecnologiche, o le discipline teoriche come matematica e fisica, ma anche le discipline ad orientamento umanistico come filosofia, linguistica, sociologia e psicologia⁴⁵.

Per le sue *caratteristiche* di determinismo e finitezza, un algoritmo si presta ad essere automatizzato, cioè ad essere eseguito da una macchina (elaboratore elettronico, computer) opportunamente programmata.

Il legame tra algoritmica e informatica è evidenziato dalla denominazione inglese di *computer science*, o scienza della computazione, mentre la definizione italiana preferisce la dicitura scienza dell'*informazione*, o scienza dei processi di informazione.

1.1. CENNI STORICI

La parola *algoritmo* deriva dal nome del matematico persiano Muhammad Al-Khwarizmi, vissuto nel IX sec. d.C.; a lui si attribuisce la formulazione di regole passo-a-passo per le operazioni aritmetiche di addizione, sottrazione, moltiplicazione e divisione di due numeri naturali. Il suo nome scritto in latino divenne *Algorismus*, da cui l'odierno etimo.

Una delle prime macchine in grado di portare avanti un processo di tipo algoritmico fu un telaio inventato dal francese Joseph Jacquard⁴⁶: il motivo da

⁴⁵Si veda Harel e Feldman (2008), cap. 1.

⁴⁶ Joseph Marie Jacquard (Lione 1752 - Oullins 1854), inventore del telaio automatico conosciuto come Telaio Jacquard, che costituisce la prima applicazione pratica delle schede perforate che guidano la realizzazione di disegni e trame sui tessuti.

tessere veniva determinato mediante schede perforate, anticipatrici delle schede di programmazione impiegate negli elaboratori a metà del XX secolo.

Il matematico inglese Charles Babbage⁴⁷ progettò e costruì varie macchine che operavano in base a procedimenti di tipo algoritmico, e in particolare disegnò il progetto di una *macchina analitica* che non doveva eseguire compiti specifici di calcolo, ma programmi codificati in base a schede perforate, quindi un equivalente matematico del telaio di Jacquard a cui si ispirò. Le idee presenti in tale modello, sebbene esso fosse di natura meccanica, formarono la base per la progettazione della struttura interna dei calcolatori elettronici.

L'ingegnere americano Herman Hollerith⁴⁸ inventò a sua volta una macchina basata su schede perforate che venne impiegata per la stesura dei tabulati del censimento del 1890.

I primi veri computer, tuttavia, appaiono solo quasi dopo mezzo secolo, alla fine della II Guerra Mondiale, e alla loro realizzazione concreta contribuirono la necessità di gestire l'informazione in modo veloce e automatizzato e di "infrangere" codici i cui contenuti erano vitali per lo sforzo bellico.

Gli studi sugli algoritmi hanno dato origine alla teoria della computabilità o della computabilità effettiva, o teoria della ricorsività. La teoria della computabilità è stata sviluppata nella prima metà del XX secolo da matematici e logici come Alan Turing⁴⁹, John von Neuman⁵⁰, Andrei A. Markov⁵¹,

⁴⁷ Charles Babbage (Londra 1791-1871) matematico e filosofo britannico, importante nella storia dell'informatica grazie alle sue macchine: della prima, la *macchina differenziale*, fu realizzato un prototipo imperfetto mentre la seconda, la *macchina analitica*, fu solo progettata. Parti dei meccanismi progettati da Babbage si trovano al Museo della scienza di Londra.

⁴⁸ Herman Hollerith (Buffalo 1860 - Washington D.C. 1929), ingegnere statunitense, lavorò per il censimento dal 1890 fino al 1896, anno in cui fondò la *Tabulating Machine Company* che, dopo varie evoluzioni, diventò nel 1924 la *International Business Machines Corporation*, meglio nota come IBM.

⁴⁹ Alan M. Turing (Londra 1912 - Wilmslow 1954), logico e matematico britannico, è considerato uno dei padri dell'informatica e uno dei più grandi matematici del XX secolo. E' famoso, oltre che per la ideazione della Macchina di Turing, su cui torneremo nel prossimo capitolo, per l'applicazione delle sue competenze di crittografia alla decodifica dei messaggi militari durante la II Guerra Mondiale.

⁵⁰ John von Neumann (Budapest 1903 - Washington 1957), matematico ungherese di origine ebraica, trasferitosi intorno al 1930 negli Stati Uniti; ha dato contributi significativi e innovativi in varie aree scientifiche, dalla matematica, alla meccanica quantistica, alla cibernetica, alla teoria dei giochi e all'intelligenza artificiale.

⁵¹ Andrej Andreevič Markov (Rjazan 1856 - San Pietroburgo 1922), matematico russo, contribuì allo sviluppo della teoria dei numeri, dell'analisi matematica, del calcolo infinitesimale, della teoria della probabilità e alla statistica, con particolare riferimento allo studio dei processi stocastici che da lui prendono il nome.

Alonzo Church⁵², Emil Post⁵³, dando origine all'area di studi nota come *informatica teorica (theoretical computer science)*⁵⁴.

1.2 ALGORITMI

Gli algoritmi possono essere definiti come metodi per la soluzione di problemi, dove un problema consiste nei dati che vengono forniti all'inizio del procedimento e nel risultato, o nei risultati, che si vogliono ottenere⁵⁵. I dati forniti in ingresso costituiscono l'*input* dell'algoritmo, mentre il risultato, ovvero i valori in uscita, rappresentano l'*output*.

Come mostra la seguente figura, possiamo rappresentare un algoritmo come la “processazione” dell'insieme dei dati in ingresso (*input*), tramite l'applicazione di una serie di operazioni all'interno della scatola, per ottenere il risultato in uscita (*output*):



FIGURA 6. *Schema di algoritmo*

Un algoritmo è un metodo che consente di calcolare il risultato desiderato a partire dai dati di partenza. Perché un procedimento costituisca un algoritmo deve essere totalmente esplicito, vanno specificati in modo preciso tutti i passi del procedimento da eseguire per ottenere il risultato in output a partire dai dati in input. Si richiede, quindi, che venga data una lista dettagliata di tutte le istruzioni e dei passaggi necessari per eseguirle.

In altre parole, un algoritmo è un procedimento di calcolo, o computazione, costituito da un insieme *finito* di istruzioni, che viene portato a termine in un numero *finito* di passi. A loro volta, le istruzioni impiegano un insieme *finito* di operazioni elementari, che si assumono come note o primitive.

⁵² Alonzo Church (Washington 1903 – Hudson 1995), matematico e logico statunitense, ha contribuito alla fondazione dell'informatica teorica ed è noto per i suoi studi sul *lambda calcolo* e sui problemi di decidibilità.

⁵³ Emil Leon Post (Augustow 1897 – New York 1954), logico e matematico statunitense, contribuì allo sviluppo dell'informatica teorica con i suoi studi sulla computabilità, la ricorsività e la completezza della logica.

⁵⁴ Per altre informazioni si rimanda a Harel e Feldman (2008); si veda anche Numerico (2005).

⁵⁵ Si vedano Frixione e Palladino (2004), Harel e Feldman (2008).

Possiamo sintetizzare, dicendo che un algoritmo:

- è un insieme finito di istruzioni per ottenere un risultato, raggiungere uno scopo, trarre un’inferenza, ecc.;
- la soluzione (se esiste) si ottiene applicando le istruzioni un numero finito di volte;
- in ogni momento deve sempre essere specificato quale istruzione va eseguita al passo successivo;
- deve essere chiaro quando il procedimento è terminato.

Gli algoritmi sono dunque procedimenti *deterministici* (metodi effettivi) che risolvono problemi, applicando le istruzioni un numero finito di volte. Trattandosi di processi deterministici, una volta fissati i dati, il risultato ottenuto sarà sempre lo stesso.

Va precisato che il fatto che le istruzioni e il procedimento sono finiti non esclude che le possibili istanze del problema siano *infinite*, e questo risultato è assicurato dall’uso di variabili e dalla ricorsività del procedimento.

Come esempi di algoritmi si possono richiamare:

- le operazioni aritmetiche;
- il procedimento euclideo per la ricerca del massimo comun divisore di due numeri naturali non nulli;
- il metodo delle tavole di verità della logica proposizionale.

Vi sono anche algoritmi di tipo non matematico, come la decisione se una parola è un palindromo o l’ordinamento di una successione di elementi (numeri, lettere, ecc.), ad esempio, dal più piccolo al più grande. Prendiamo in considerazione l’esempio del *palindromo*, si tratta di un algoritmo con risposta **SI** o **NO**:

si inizia confrontando la prima e l’ultima lettera della parola (stringa di lettere); se la prima lettera è diversa dall’ultima il procedimento termina, e la risposta è negativa; se le due lettere sono uguali, si cancellano e si riprende il controllo delle due lettere successive. Se, dopo un certo numero di iterazioni del procedimento non è rimasta alcuna lettera, ovvero se la sequenza di lettere si cancella a 0, allora la parola di partenza è un palindromo⁵⁶ (si veda la FIGURA 7).

⁵⁶ Esempi di palindromi con parole più lunghe di quattro lettere sono rari; fra i sostantivi di sette lettere troviamo: *anilina*, *ossesso* e *ingegni*; di otto lettere: l’arcaismo di *ereggere*; di nove lettere la voce verbale *onorarono*. Al di sopra delle nove lettere si trovano solo parole inventate per oggetti impossibili.

E' interessante anche l'algoritmo euclideo⁵⁷ delle divisioni successive come metodo per determinare il *Massimo Comun Divisore* (MCD) di due numeri naturali non nulli, a e b : l'input dell'algoritmo sono i due numeri, l'output è il loro MCD. Tale metodo si basa sul fatto che ogni numero che divide il *dividendo* e il *divisore*, divide anche il *resto* di una divisione. Ad esempio:

Determiniamo MCD di 2.079 e 987

- dividiamo 2.079 per 987 → otteniamo 2 con resto 105
- dividiamo 987 per 105 → otteniamo 9 con resto 42
- dividiamo 105 per 42 → otteniamo 2 con resto 21
- dividiamo 42 per 21 → otteniamo 2 con resto 0

Il procedimento può essere così formulato: dati due numeri naturali a e b , si divide a per b trovando il resto r ; se r è uguale a 0, allora b è il MCD cercato; altrimenti si divide il resto r per b e si ottiene il resto p ; si procede ripetutamente finché si raggiunge il risultato 0.

Molti algoritmi forniscono una risposta di tipo SI e NO, si tratta di algoritmi di decisione; vi sono tuttavia altri tipi di algoritmi che possono aver come output numeri, proposizioni, ecc. Ad esempio, l'algoritmo che risolve il problema di cercare un numero di telefono nella guida telefonica: in questo caso l'input è rappresentato (i) dalla guida, ovvero dall'elenco dei nomi degli utenti posti in ordine alfabetico accompagnati dal rispettivo numero telefonico e (ii) dal nome della persona di cui si sta cercando il numero; l'output sarà rappresentato dal numero di telefono di questa persona. Se il nome non è sulla guida, la ricerca fallisce.

Un altro esempio è dato dall'algoritmo per determinare la somma degli stipendi degli impiegati di una azienda prendendo in input l'insieme delle cartelle che contengono il nome dell'impiegato e il suo stipendio. L'algoritmo inizierà (i) annotando il numero 0, poi (ii) procederà lungo la lista aggiungendo lo stipendio di ciascun impiegato al numero precedentemente annotato; (iii) raggiunta la fine della lista, produrrà come output il numero annotato che corrisponde alla somma di tutti gli stipendi⁵⁸.

2. COSTRUZIONE DI GRAFI PER ALGORITMI

Una rappresentazione naturale di un algoritmo è quella ottenuta mediante grafi con nodi per indicare le operazioni che vengono eseguite e archi per

⁵⁷ Euclide (323 a.C. - 285 a.C.) descrisse questo algoritmo nel suo libro degli *Elementi*. Invece di usare i numeri interi, tuttavia, utilizzò segmenti di retta.

⁵⁸ Per approfondimenti si rimanda a Frixione e Palladino (2004), Harel e Feldman (2008).

indicare le relazioni tra nodi, con un solo nodo di inizio e almeno un nodo di conclusione del processo. Ci soffermeremo in particolare su una tipologia di diagrammi chiamati diagrammi di flusso.

2.1 DIAGRAMMI DI FLUSSO

I *diagrammi a blocchi*, detti anche diagrammi di flusso, in inglese *flow chart*, sono dei grafi orientati i cui nodi rappresentano le istruzioni da eseguire, dove la forma di ciascun nodo è associata al tipo di istruzione da eseguire e gli archi che li collegano indicano l'ordine in cui le istruzioni devono essere eseguite. Si tratta di una modalità grafica di rappresentazione del flusso di informazione esibito da un processo o, più in generale, da un algoritmo.

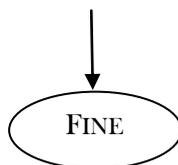
Un *flow chart* descrive i diversi passaggi o operazioni dell'algoritmo sotto forma di uno schema, o diagramma, in cui le diverse fasi del processo e le condizioni che devono essere rispettate, sono rappresentate da simboli iconici detti blocchi elementari. I blocchi (nodi del grafo) sono collegati tra loro da frecce (archi del grafo) che indicano le tappe, i passaggi, in base a cui si sviluppa il procedimento.

Un flow chart presenta diversi tipi di blocchi semplici o elementari; in primo luogo il blocco di *inizio*, da cui parte una freccia, e il blocco di *fine* del processo, al quale arriva l'ultima freccia:

BLOCCO INIZIALE

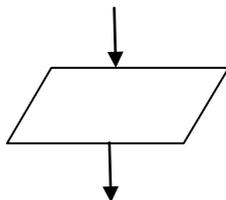


BLOCCO FINALE

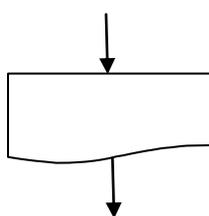


Vi sono poi blocchi che rappresentano le operazioni di input e output:

BLOCCO DI INPUT

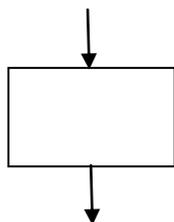


BLOCCO DI OUTPUT

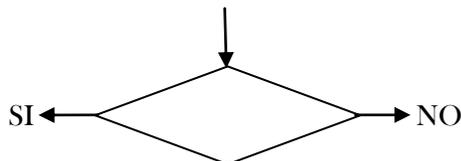


e blocchi che descrivono operazioni o interrogazioni (blocchi di scelta):

BLOCCO DI ISTRUZIONE

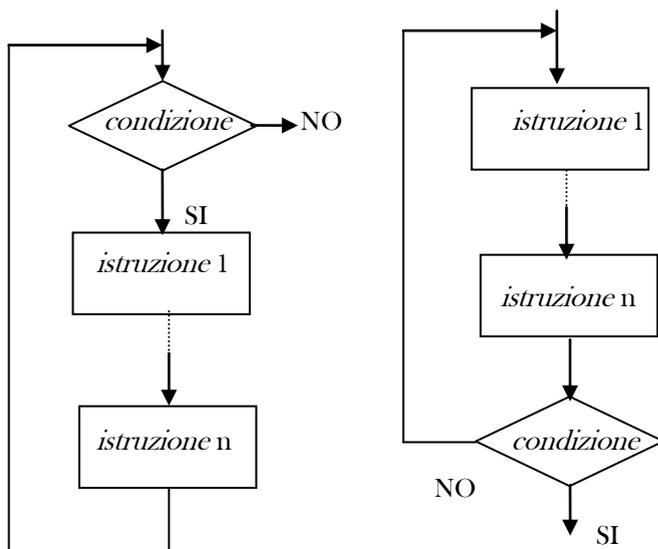


BLOCCO TEST



I nodi del diagramma a forma di rettangolo contengono le istruzioni, o elaborazioni, relative alle operazioni da compiere e possono occorrere anche in successione; i nodi a forma di rombo contengono le condizioni del test che si applica nel procedimento: una espressione è un test se assume i valori di verità vero (freccia SI) o falso (freccia NO).

I blocchi di test, chiamati anche blocchi di controllo, possono essere impiegati per la definizione di *cicli*, mediante i quali una stessa istruzione o un gruppo di istruzioni possono essere ripetuti più volte. I seguenti sono due esempi:



Vi sono algoritmi che per certi input non producono alcun risultato perchè danno origine ad un calcolo che non termina. In questo caso si dice che

l'algoritmo va in *loop* o entra in un circolo vizioso. Ad esempio, si prenda in input il valore x di un qualsiasi numero maggiore di 100; la condizione $x \neq 100$ risulterà vera, quindi si aumenta di 1 il valore della variabile x e si torna a verificare la condizione; poiché la condizione risulterà sempre vera, l'algoritmo continuerà a girare su sé stesso.

I diagrammi a blocchi devono rispettare una serie di condizioni rilevanti per la loro buona formazione e, quindi, il loro funzionamento. Precisamente, una combinazione di blocchi elementari descrive un algoritmo se:

- viene usato un numero finito di blocchi
- lo schema inizia con un *blocco iniziale* e termina con un *blocco finale*
- ogni blocco soddisfa delle *condizioni di validità*

I blocchi devono poi osservare le seguenti condizioni di validità (buona formazione):

- *blocco azione* (input- output) e *blocco istruzione*: ogni blocco di questi due tipi ha una sola freccia entrante e una sola freccia uscente
- *blocco test o di controllo*: ogni blocco di questo genere ha una sola freccia entrante e due frecce uscenti
- condizioni sulle frecce: ogni *freccia* deve entrare in un blocco, o uscire da un blocco

Sulla base di queste condizioni, i diagrammi di flusso svilupperanno precise architetture, come dimostrato dal *Teorema fondamentale della programmazione strutturata*⁵⁹ che asserisce ogni algoritmo (programma) può essere codificato nei termini esclusivamente di tre strutture fondamentali: (i) *sequenziale*, quando il diagramma si sviluppa in modo lineare secondo una semplice elencazione di istruzioni; (ii) *iterativa*, quando il diagramma presenta cicli che fanno tornare il programma su sé stesso (vedi FIGURA 8); (iii) *condizionale*, quando il diagramma si sviluppa secondo due condizioni (di tipo se ... allora) a destra e a sinistra di un blocco di controllo (vedi FIGURA 7).

2.2 ESEMPI DI DIAGRAMMI DI FLUSSO

Consideriamo ora alcuni esempi di algoritmi e i diagrammi di flusso introdotti per rappresentarli.

⁵⁹ Jacopini e Böhm, 1966.

Il seguente è un esempio di diagramma di flusso che rappresenta l'algoritmo per determinare se una parola, come le seguenti, aBBa, esse, ossesso, emme, Ada, Anna, onorarono, è un palindromo⁶⁰:

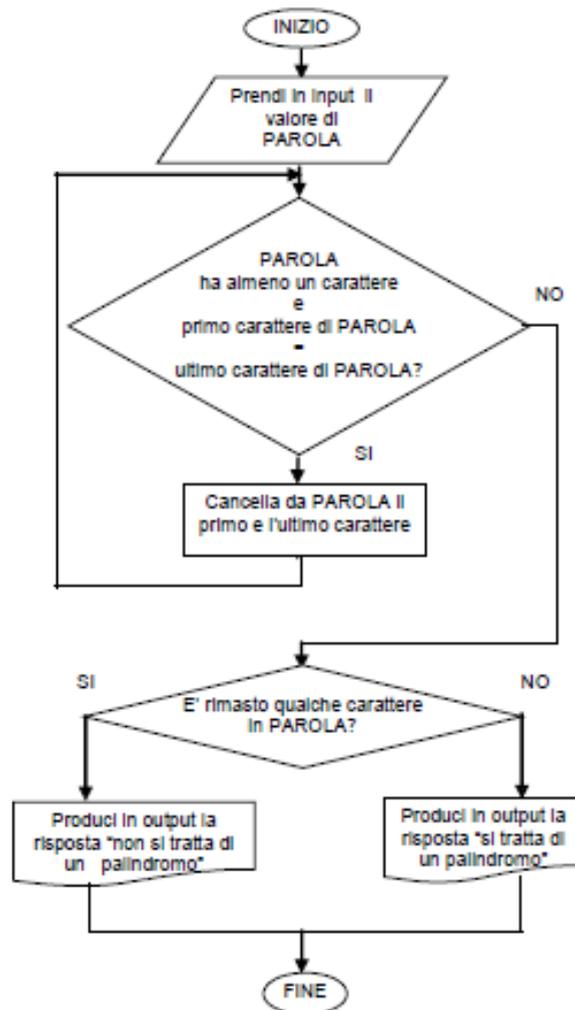


FIGURA 7: Algoritmo per riconoscere un palindromo

⁶⁰ Si veda Frixione e Palladino (2004), p.32.

Con il seguente diagramma di flusso possiamo rappresentare il percorso logico della istruzione: posso fumare che abbiamo visto a pag.

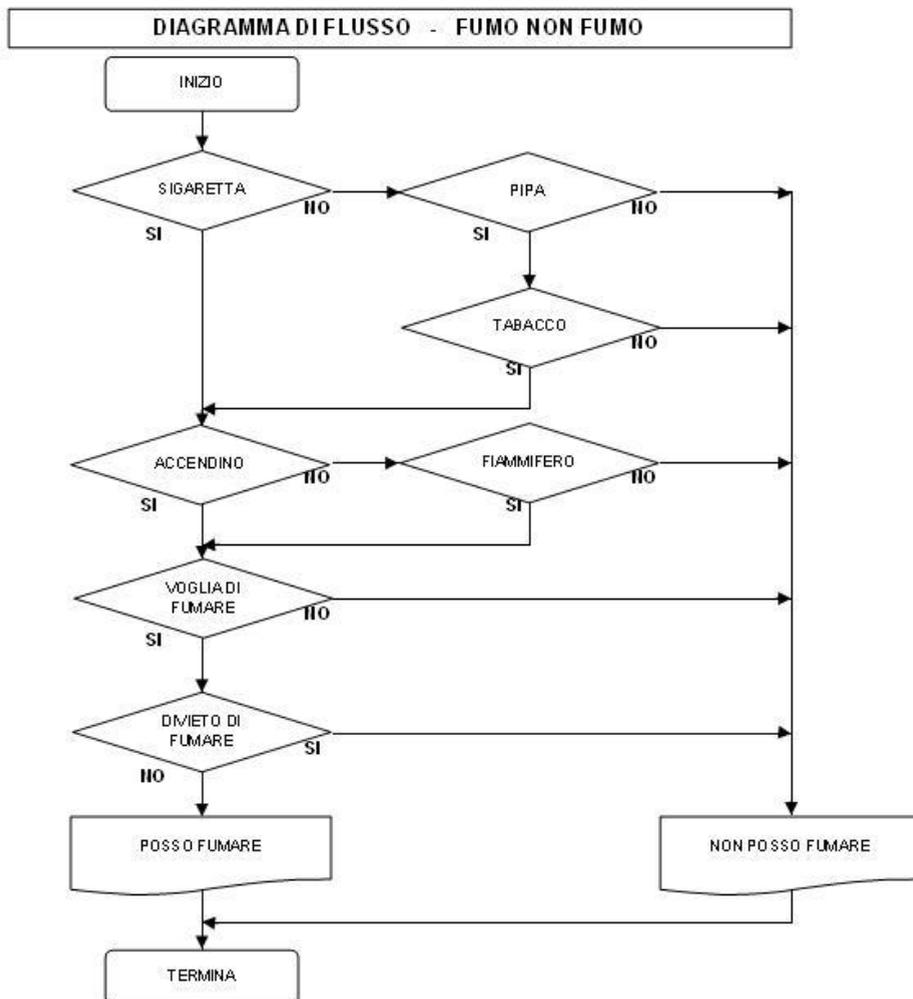


FIGURA 8: *Algoritmo per "posso fumare"*

Il procedimento per la creazione di un diagramma di flusso, al fine di risolvere logicamente un problema, può essere illustrato dal seguente esempio relativo all'ordinamento di una sequenza di numeri.

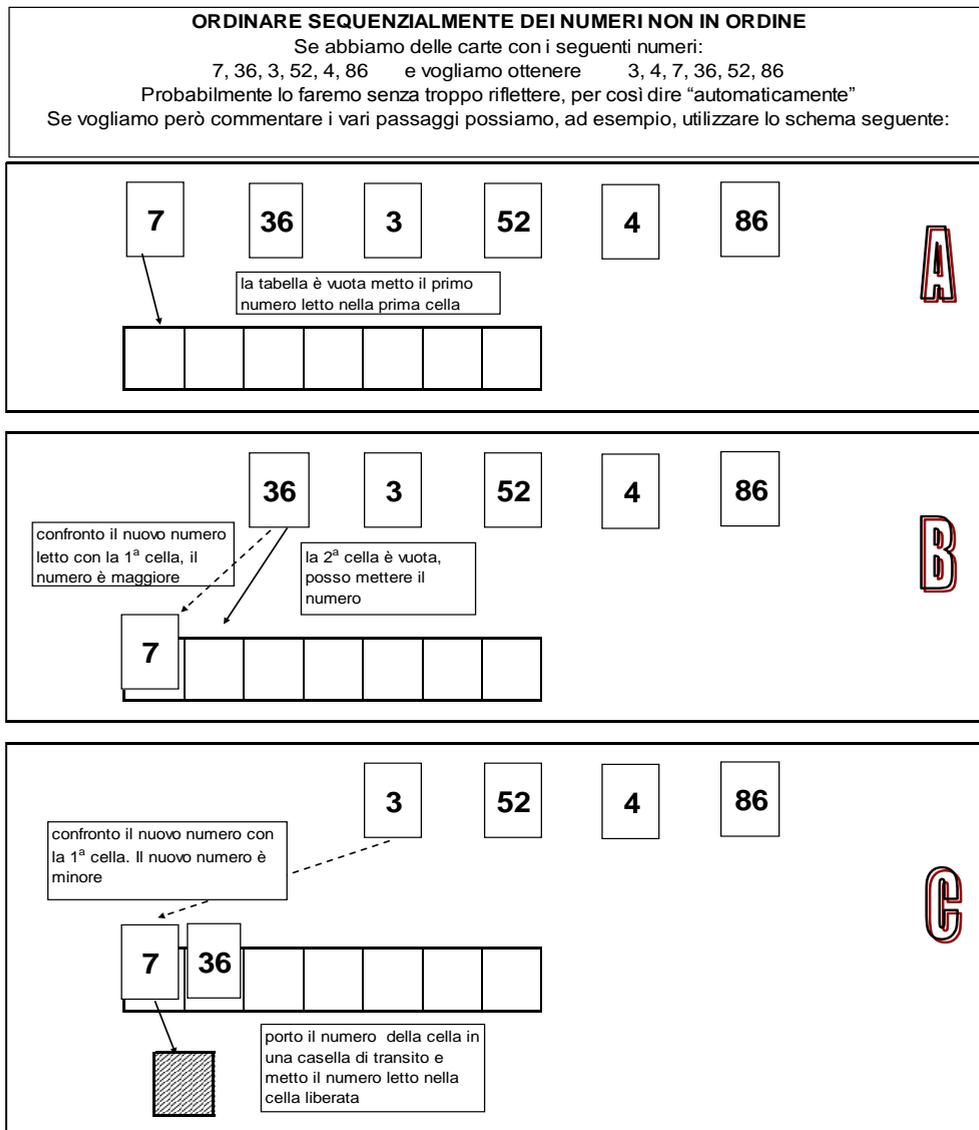


FIGURA 9

Nella prima parte del procedimento si prende una sequenza non ordinata di numeri e una tabella in cui inserirli in modo che vengano scritti in ordine dal minore al maggiore. Si inizia individuando il primo numero della sequenza e inserendolo nella prima cella della tabella vuota (fase A); si prende poi il secondo numero e lo si confronta con il primo; se è maggiore lo si inserisce nella seconda cella (fase B). Si prende il terzo numero, lo si confronta con il primo e, nel caso sia minore, si sposta il primo numero in una casella di transito e si inserisce il terzo numero al suo posto nella prima cella che si è liberata (fase C). Quindi si confronta il numero nella casella di transito con il secondo numero, se è minore di questo, lo si mette al suo posto e si sposta il secondo numero nella terza cella vuota (fase D). Ripeto queste operazioni per gli altri numeri della sequenza (fasi E, F, G).

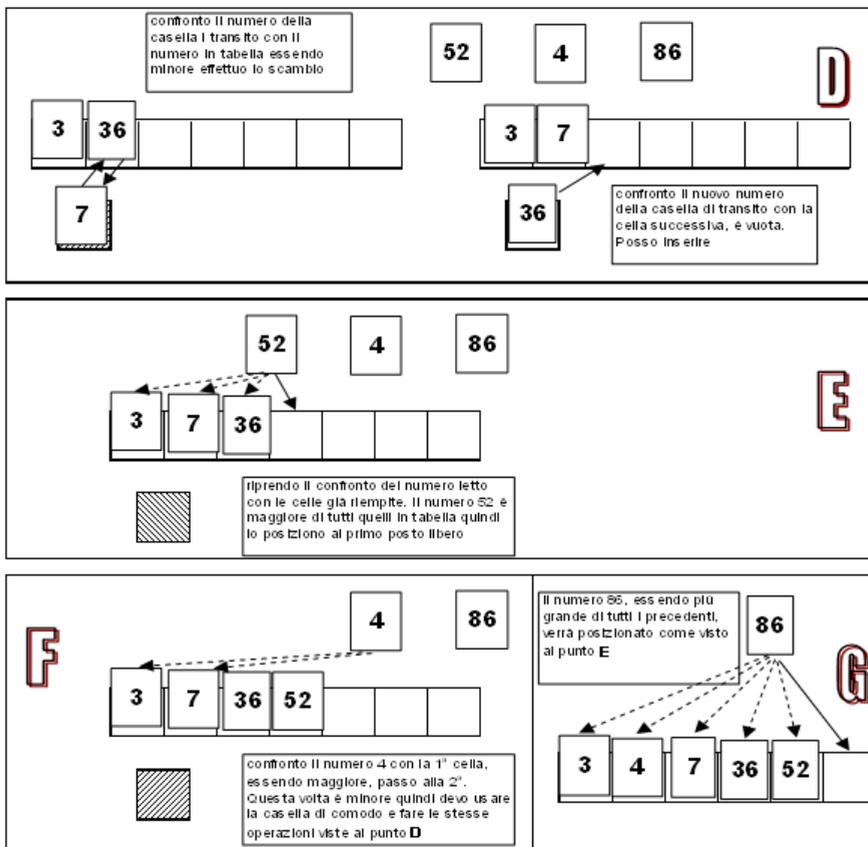


FIGURA 10

Il procedimento può essere rappresentato nei suoi elementi essenziali dal seguente diagramma di flusso:

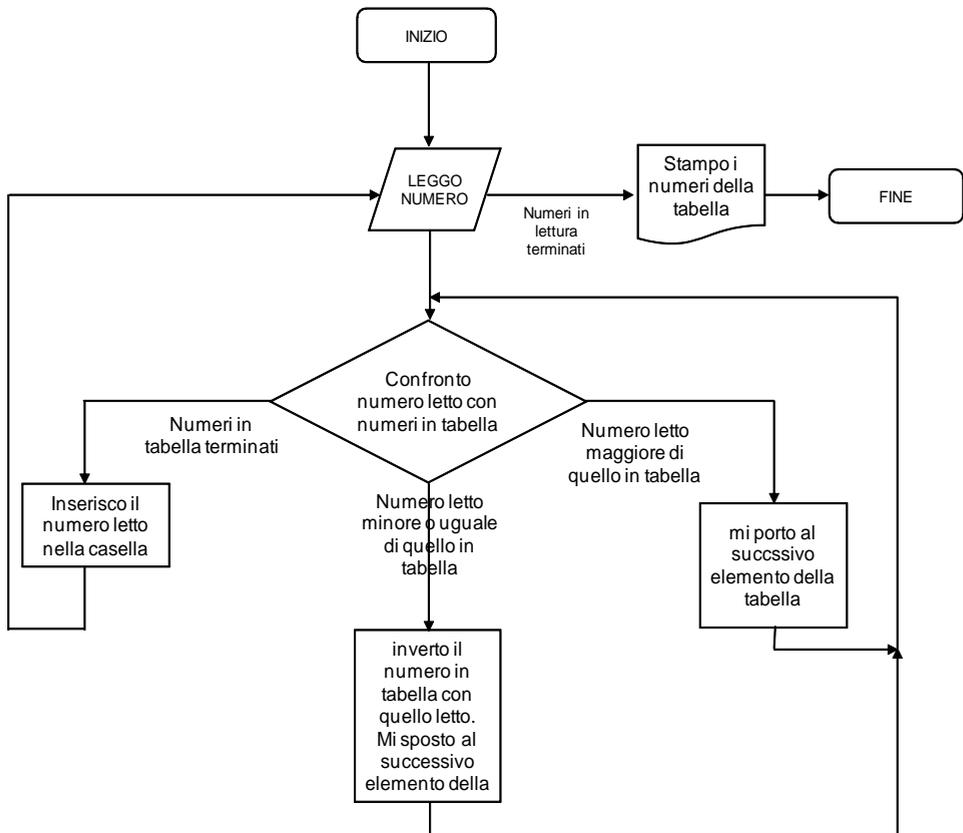


FIGURA 11

Questo schema, certamente sufficiente a spiegare i diversi passaggi logici che sono stati fatti per ordinare la sequenza di numeri, non è però sufficiente ad “istruire” (programmare) un elaboratore affinché possa mettere in sequenza ordinata dei numeri. Per poterlo fare dovremo aggiungere molti altri dettagli alla *flow chart* come le posizioni dei vari numeri nella sequenza, il loro posizionamento nelle celle, i passaggi richiesti per confrontarli con gli altri numeri, e così via.

2.3 RICORSIVITÀ

Con questo termine si intende la capacità di una procedura di richiamare sé stessa. È uno degli aspetti più utili e interessanti dei procedimenti di calcolo e in particolare degli algoritmi. Ad esempio, i costrutti iterativi, in cui uno stesso testo può corrispondere a varie parti del processo che descrive, consentono di creare processi lunghi a partire da testi relativamente brevi.

Un esempio naturale di ricorsione è rappresentato dal procedimento di risoluzione del rompicapo delle Torri di Hanoi⁶¹.

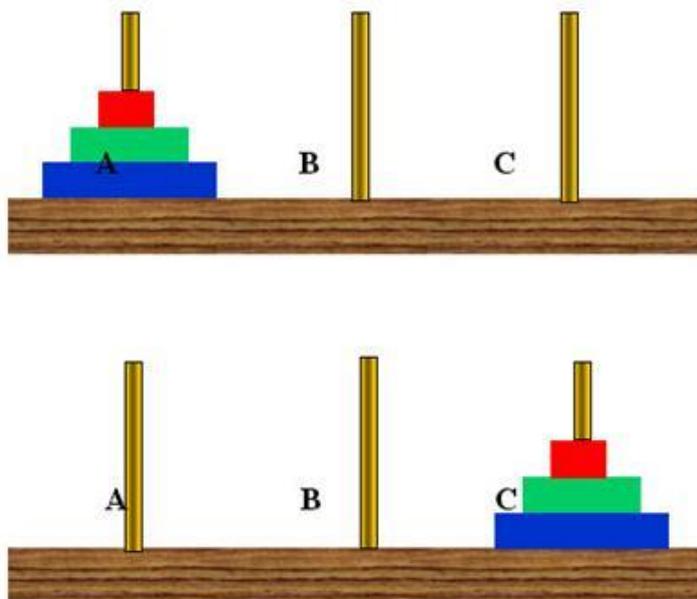


FIGURA 12: *Torri di Hanoi con tre dischi*

Indichiamo con A, B e C le tre colonnine. Nel caso banale di un unico disco, occorrerà un solo movimento per risolvere il gioco, basta infatti spostare il disco dalla colonnina A alla colonnina C. Con due dischi, sono necessari 3 movimenti, si deve spostare il disco superiore sulla colonnina B, il disco più grande su C ed infine l'altro disco sempre su C.

⁶¹ Il gioco venne inventato nell'Ottocento da Edouard Lucas, studioso di teoria dei numeri, che deve la sua fama all'analisi della successione di Fibonacci.

Quali sono gli spostamenti minimi necessari per trasferire la torre a tre dischi da A a C? Dobbiamo spostare il disco superiore su C e quello di mezzo su B, sul quale spostiamo poi il disco più piccolo. In seguito, spostiamo il disco più grande su C, quello più piccolo su A e, per finire, il disco da B a C e da A a C.

Abbiamo, in totale, sette movimenti. Con un po' di pratica si arriva facilmente a capire il procedimento da seguire con un numero qualsiasi di dischi, scoprendo la formula risolutiva del gioco. Possiamo così esemplificare l'algoritmo delle Torri di Hanoi, indicando con M il numero delle mosse e con n il numero dei dischi:

Algoritmo:

Dischi (n) : 1 2 3 4 5 6 7 8 ...

Mosse (M) : 1 3 7 15 31 63 127 255...

$$M(1) = 2^1 - 1 = 1.$$

$$M(n) = 2^n - 1:$$

$$2^2 - 1 = 3; 2^3 - 1 = 7; 2^4 - 1 = 15; 2^5 - 1 = 31; 2^6 - 1 = 63; 2^7 - 1 = 127; \text{ecc.}$$

La ricorsività è un fenomeno che si manifesta in vari ambiti della esperienza umana; ad esempio nelle lingue naturali incontriamo diverse costruzioni in cui una regola linguistica può essere applicata al risultato di una sua precedente applicazione⁶².

Un esempio tipico di ricorsività linguistica è rappresentato dalla generazione di *clausole relative* come specificazioni degli argomenti di una frase semplice:

- 1) Maria mangia il dolce.
 - 2) Maria mangia il dolce che è sul tavolo.
 - 3) Maria mangia il dolce che è sul tavolo, che si trova in cucina.
- ecc.

Come si vede, le clausole relative subordinate possono agganciarsi a segmenti di frase, che sono a loro volta subordinati alla frase principale. In queste situazioni, il numero di agganci o subordinazioni che possono essere accettati in un discorso è direttamente correlato al carico sostenuto dalla memoria di lavoro dei parlanti⁶³.

⁶² In questo senso, la semiotica parla di *codici non ricorsivi*, come certi comportamenti animali (p. es. la danza delle api), e *codici ricorsivi* come le lingue umane o i sistemi matematici. Il problema della ricorsività di certi costrutti delle lingue naturali, come le clausole relative dell'esempio, ha svolto un ruolo cruciale nello sviluppo della teoria trasformazionale della sintassi di Noam Chomsky (1957, 1965).

⁶³ Si richiamano in proposito De Vincenzi (1989) e (1991).

ESERCIZI

E1 DIAGRAMMI DI FLUSSO

ESERCIZIO 1:

Fornire esempi di diagrammi di flusso, richiamandosi alle figura di pag.14.

ESERCIZIO 2:

Fornire esempi di inferenze logiche mediante il metodo dei diagrammi di flusso.

ESERCIZIO 3:

Riflettere sull'importanza dell'impiego di immagini dinamiche per rappresentare il "flusso dell'informazione", fornire una spiegazione dal punto di vista psicologico e cognitivo.

E2 ALGORITMI

ESERCIZIO 1:

Fornire una rappresentazione dell'algoritmo per determinare la somma degli stipendi degli impiegati descritto negli esercizi. Provare a costruire il corrispondente diagramma di flusso.

ESERCIZIO 2:

Trovare processi dell'esperienza umana che possono essere considerati come algoritmi e fornirne una rappresentazione.

ESERCIZIO 3:

Provare a costruire un diagramma di flusso completo per l'esempio di ordinamento di una sequenza di numeri

E3 RICORSIVITÀ

ESERCIZIO 1:

Trovare esempi di procedimenti ricorsivi in ambito logico e matematico.

ESERCIZIO 2:

Trovare esempi di procedimenti ricorsivi in ambito linguistico.

CAPITOLO QUARTO

AUTOMAZIONE

1. CENNI SULLA NASCITA DELLA CIBERNETICA

È opportuno aprire questo capitolo con alcune osservazioni di carattere storico⁶⁴. Sono molte le analogie tra mondo della natura e mondo della tecnica che la filosofia e il pensiero scientifico hanno messo in evidenza, spesso facendo ricorso a suggestive metafore che sono servite a favorire la comprensione delle conquiste umane a partire dalla conoscenza posseduta del contesto naturale e sociale. Si consideri, ad esempio, la similitudine delineata da Cartesio⁶⁵ tra dispositivi di tipo elementare (meccanico, idraulico o pneumatico) e la costruzione di modelli per la rappresentazione delle funzioni animali. Oppure l'estensione all'organismo umano operata da La Mettrie⁶⁶ dell'idea cartesiana di similitudine tra macchine artigianali e corpi della natura. Una metafora che ha avuto particolare fortuna in questo senso è quella del sistema circolatorio animale come sistema idraulico: *la pompa del cuore, i condotti venosi, ecc.*

Nello sviluppo del pensiero scientifico e dei risultati tecnologici la cultura umana viene assunta come modello del funzionamento delle strutture naturali, e viceversa. Da un lato, i prodotti dell'evoluzione della natura sono considerati come prototipi dell'evoluzione culturale umana, dall'altro, i prodotti della cultura diventano i risultati dei processi di evoluzione.

In questo quadro si pone anche lo sviluppo della scienza dell'automazione e dell'intelligenza artificiale che viene chiamata *cibernetica*, la cui etimologia proviene dalla parola greca *kybernetiké (téchné)*, che significa arte di *condurre, guidare*, propria del pilota o del timoniere. Il termine fu impiegato

⁶⁴ Per un quadro più esaustivo si rimanda al saggio *Indagini meccanicistiche sulla mente: la cibernetica e l'intelligenza artificiale*, Introduzione a Somenzi e Cordeschi (1986), pp. 19-61; si veda anche Numerico (2005).

⁶⁵ René Descartes (La Haye, Touraine, 1596 - Stoccolma 1650), italianizzato come Cartesio, ha contribuito in modo determinante alla fondazione della filosofia razionalista e della matematica moderna, ponendo le basi della concezione *dualistica* del rapporto mente e corpo (si vedano *Opere* e *Discorso sul metodo*).

⁶⁶ La Mettrie J. O. de (Saint-Malo 1709 - Berlino 1751), medico e filosofo francese sostenne la stretta interdipendenza tra anima e corpo e la riconciliabilità dei processi mentali a modificazioni corporee. Nella *Storia naturale dell'anima* (1745) applica ai processi psichici (anima) le teorie cartesiane sulla natura e il funzionamento dei corpi viventi.

a partire dalla fine del 1700 per indicare l'arte di governare e regolare le macchine (p.es. la macchina a vapore di Watt), ma viene esteso nel '900, in particolare nel corso della seconda guerra mondiale, al governo dei dispositivi elettrici e in genere dei meccanismi automatici. La definizione della cibernetica come scienza che studia i messaggi e il controllo della comunicazione è dovuto a Wiener (1948)⁶⁷.

Oggi, possiamo dire che la cibernetica è la scienza che studia i fenomeni di autoregolazione e comunicazione nell'ambito dei sistemi artificiali e negli organismi naturali, in collaborazione con la teoria dell'informazione, la computer science, le scienze cognitive e gli studi sull'intelligenza (intelligenza artificiale).

1.1 PRINCIPI TEORICI

Vi sono alcuni momenti teorici che hanno avuto un ruolo decisivo per la nascita e lo sviluppo della cibernetica: (i) la caratterizzazione dell'equivalenza tra comportamento teleologico degli organismi e comportamento degli artefatti a retroazione negativa (Rosenblueth⁶⁸, Wiener e Bigelow⁶⁹); (ii) la descrizione del sistema nervoso come 'realizzatore di modelli' (Craik⁷⁰); (iii) lo studio della logica delle reti neurali (McCulloch⁷¹ e Pitts⁷²).

⁶⁷ Norbert Wiener (Columbia 1894 - Stoccolma 1964), matematico e filosofo statunitense, di origine russa, studiò a Cambridge con Russell e a Gottinga con Hilbert e Husserl; dal 1919 ha lavorato al MIT. Ha contribuito alle ricerche sul calcolo delle probabilità, ha sviluppato la *teoria dell'informazione* assieme al suo allievo Claude Shannon, ed è considerato il fondatore della cibernetica come scienza della comunicazione.

⁶⁸ A. Rosenblueth Stearns (Ciudad Guerrero 1900- Città del Messico 1970) medico e neurofisiologo messicano, ha lavorato alla Harvard Medical School ed è uno degli ispiratori della cibernetica; con Wiener e Bigelow scrisse *Behavior, Purpose and Teleology* (1943), articolo che pone le basi della nuova disciplina.

⁶⁹ Julian H. Bigelow (Nutley 1913 - Princeton 2003) pioniere nel campo dell'ingegneria informatica, collaborò con Wiener alle ricerche sul controllo automatico delle macchine, in particolare durante la seconda guerra mondiale, e con von Neuman alla costruzione di uno dei primi computer digitali (macchina IAS) presso l'Institute for Advanced Studies di Princeton.

⁷⁰ Kenneth J. W. Craik (1914 - 1945) filosofo e psicologo britannico, oltre alle ricerche di psicologia sperimentale, studiò le proprietà dei sistemi automatici di controllo dell'informazione, applicandoli all'organizzazione della mente; si veda Craik (1943).

⁷¹ Warren S. McCulloch (Orange N. J. 1898 - Cambridge 1969), neurofisiologo statunitense, contribuì allo sviluppo della cibernetica con le sue ricerche sulla organizzazione del cervello e le reti neurali; lavorò dal 1952 al MIT; si veda McCulloch & Pitts (1943), *A logical calculus of the ideas immanent in nervous activity*.

⁷² Walter Harry Pitts (Detroit 1923 - Cambridge, Mass 1969), logico statunitense, ha contribuito allo sviluppo della psicologia cognitiva con le sue ricerche sui processi e le reti

Wiener definisce la cibernetica come una disciplina che studia in un'unica teoria e con metodo unitario, tanto i meccanismi fisiologici che gli automi. Si tratta di una scienza ad orientamento fortemente interdisciplinare che si occupa non solo del controllo automatico dei macchinari mediante appropriati strumenti elettronici (i predecessori degli attuali computer), ma anche dello studio del cervello umano, del sistema nervoso e del rapporto tra i due sistemi di comunicazione e di controllo, artificiale e biologico⁷³.

Le ricerche sull'automazione in questo periodo chiamano in causa due fondamentali concetti teorici: l'idea di modello *servomeccanico* e il principio di *retroazione*. Il termine 'servomeccanismo' è stato coniato per designare macchine che hanno un comportamento intrinsecamente rivolto ad uno scopo (*purpose*), o *teleologico*⁷⁴; questo tipo di comportamento è caratterizzato dalla proprietà detta della retroazione (o *feed-back*). La retroazione può essere definita come la capacità di un sistema dinamico di tenere conto dei risultati delle proprie operazioni, al fine di modificare le caratteristiche del sistema stesso. Si possono avere due tipi di retroazione⁷⁵:

— RETROAZIONE POSITIVA

i risultati del sistema vanno ad amplificare il funzionamento del sistema stesso: parte dell'energia in uscita della macchina viene restituita in ingresso. Poiché la retroazione positiva amplifica i segnali in ingresso, ma non li corregge, i sistemi con retroazione positiva sono instabili e portano il sistema a divergere.

— RETROAZIONE NEGATIVA

il comportamento di un oggetto (macchina, organismo in senso lato) è regolato dal margine di errore in cui l'oggetto si trova ad un dato tempo, rispetto ad un obiettivo specifico; la retroazione è *negativa* perché i segnali dell'obiettivo sono

neurali che hanno influenzato sia la cibernetica che le neuroscienze e le scienze cognitive. Ha fatto parte del *Research Laboratory of Electronics* del MIT dal 1952 al 1969.

⁷³ Cfr. Wiener (1948). Anche Craik si proponeva di esplorare sistematicamente la possibilità di applicare il principio di retroazione allo studio dell'apprendimento e del comportamento teleologico adattivo; cfr. Craik (1947). L'idea di individuare negli esseri biologici e nei dispositivi meccanici lo stesso comportamento adattivo ha costituito il punto di partenza delle ricerche interdisciplinari che nella prima metà del 1900 hanno visto lavorare fianco a fianco matematici, ingegneri elettronici, fisici, fisiologi, psicologi, biologi nella costruzione di modelli del comportamento che replicassero il funzionamento dei servomeccanismi. Sono queste ricerche che hanno portato alla fondazione delle contemporanee scienze computazionali, di intelligenza artificiale e robotica.

⁷⁴ Si veda in proposito il saggio *Behavior, purpose and teleology*, di Rosenblueth, Wiener e Binglew (1943), considerato uno dei manifesti della cibernetica.

⁷⁵ Un esempio di retroazione positiva è dato da un microfono che viene amplificato da un altoparlante, producendo un effetto sonoro crescente; un esempio di retroazione negativa è dato dal termostato che regola il riscaldamento domestico, adattandosi alle variazioni di temperatura.

usati per limitare i risultati, che altrimenti supererebbero l'obiettivo. In questo senso, i risultati (output) del sistema inibiscono il funzionamento del sistema stesso, stabilizzandolo. I sistemi con retroazione negativa sono in genere stabili e portano il sistema a convergere.

Una *macchina* con retroazione ha la caratteristica di (i) consentire uno scambio con l'ambiente e (ii) fornire una informazione regolata in vista di uno scopo, e può essere rappresentata da questo schema:

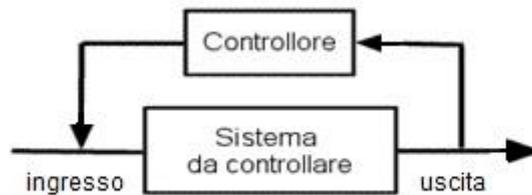


FIGURA 13: *Schema a blocchi di un controllo in retroazione*

1.2 CIBERNETICA E COMUNICAZIONE

Il ruolo fondamentale assolto da queste idee, non solo in ambito tecnologico, ma soprattutto a livello sociale e culturale, nella delineazione di una prospettiva futura in cui l'automazione sarebbe stata al servizio dell'umanità, è evidenziato con convinzione negli scritti degli autori che hanno contribuito allo sviluppo della cibernetica, in primo luogo Norbert Wiener⁷⁶.

Secondo Wiener “la vita sociale dell'uomo può essere compresa soltanto attraverso lo studio dei messaggi e dei mezzi di comunicazione relativi ad essi: messaggi fra l'uomo e le macchine, fra le macchine e l'uomo, fra macchine e macchine” Non si tratta di meccanismi predisposti unicamente a svolgere la propria particolare funzione interna, come ad esempio il carillon di un orologio, ma di macchine in grado di interagire con il mondo esterno:

“Le vecchie macchine e specie i primi tentativi di costruire gli automi, erano basate sul principio ... del meccanismo di orologeria. Le macchine moderne, invece, sono provviste di organi sensori, cioè organi di ricezione dei messaggi che provengono dall'esterno. Esse possono essere semplici come una cellula fotoelettrica, che modifica il suo comportamento non appena la luce cade su di essa, e sa distinguere la luce dall'oscurità, oppure complesse come un ricevitore televisivo.”⁷⁷

⁷⁶ Wiener (1950), trad. it. pp. 23-24. Un quadro efficace e dettagliato di queste posizioni è offerto dalla antologia di Somenzi e Cordeschi (1986).

⁷⁷ Ibid., p. 25

Il passaggio dalla messa in funzione di una macchina che agisce sul mondo esterno in base a certi messaggi (ad esempio il dispositivo elettronico di apertura di una porta automatica), alla esecuzione effettiva del suo compito (l'apertura della porta), può essere semplice o avere vari gradi complessità. In una azione complessa, la combinazione dei dati introdotti (*immissione* o *segnale*) al fine di ottenere un certo effetto sul mondo esterno (*emissione* o *risposta*), può richiedere un numero molto grande di operazioni: i dati immessi al momento, le registrazioni relative a quei dati, precedentemente immagazzinate in quella che viene chiamata la *memoria* della macchina. Per l'uso fatto all'epoca di nastri magnetici o perforati per la registrazione, i dati immessi nell'elaborazione dell'informazione vengono chiamati col termine *nastro*. Per una esemplificazione, si consideri il seguente schema⁷⁸:

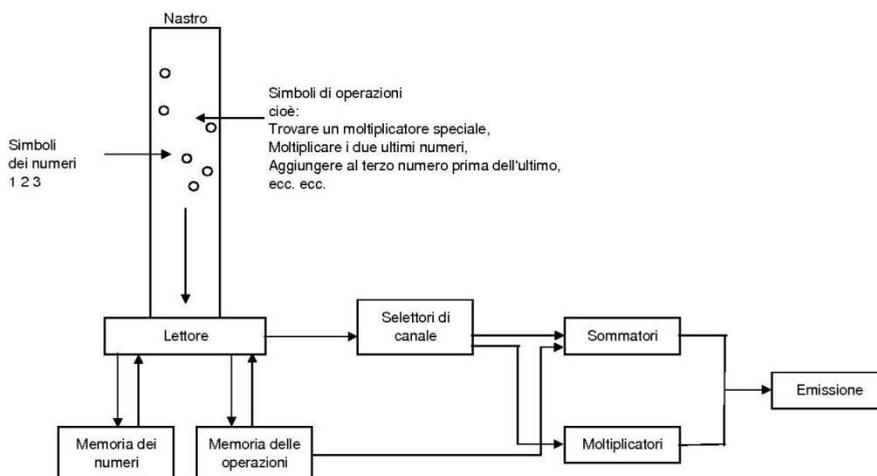


FIGURA 14: *Esempio di sistema di immissione su nastro*

Le idee di Wiener trovano piena corrispondenza nella teoria cognitiva di Kenneth Craik in base a cui il cervello umano opererebbe in modo simile a quello delle macchine⁷⁹. Secondo Craik i meccanismi del cervello sono in grado di costruire modelli interni del mondo e l'azione è il risultato della elaborazione di tali rappresentazioni interne, per cui i fenomeni mentali

⁷⁸ Wiener (1950), trad. it. p. 27.

⁷⁹ Craik (1943). Come sottolineato nella Introduzione a Somenzi e Cordeschi (1986), il 1943 fu un anno che vide la convergenza di tre importanti opere: il saggio di Rosenblueth, Wiener e Bigelow sul comportamento teleologico e la retroazione, il saggio di McCulloch e Pitts sulle reti neuronali e il libro di Craik sul sistema nervoso.

dependerebbero non dalla struttura del cervello, ma dal modo in cui esso è funzionalmente organizzato. Il modello di Craik costituisce in questo senso un significativo passo avanti rispetto alla tradizione cartesiana: mentre l'automa cartesiano è in grado unicamente di rispondere agli stimoli esterni con delle azioni meccaniche, l'automa craikiano trasforma gli stimoli esterni in una rappresentazione del mondo, riflette su tale rappresentazione e poi produce l'azione⁸⁰.

Craik ha evidenziato due possibili metodi di studio del comportamento animale e umano: *analitico* e *sintetico*; il primo riguarda il comportamento anatomo-fisiologico o strutturale, è di carattere funzionale, ed è orientato allo studio della struttura del proprio oggetto; il secondo rappresenta la ricerca dei principi fondamentali di cui uomini e macchine sono esemplificazioni specifiche; si tratta di una forma di comportamentismo fondato sullo studio delle relazioni tra l'oggetto (un organismo in senso lato) e l'ambiente⁸¹.

2. INTELLIGENZA ARTIFICIALE

Artificial Intelligence is the science of making machines do things that would require intelligence if done by men.

Marvin Minsky⁸²

Does it even make sense to speak of a machine thinking or is a thinking computer a conceptual absurdity, like a grinning electron or a married bachelor?

Jack Copeland⁸³

L'espressione *Intelligenza Artificiale*, in inglese *Artificial Intelligence* (AI), fu coniata nel 1956 dal matematico americano John McCarthy⁸⁴, durante la storica conferenza interdisciplinare svoltasi al Dartmouth college, nel New Hampshire⁸⁵. Secondo le parole di Marvin Minsky, uno dei "pionieri" della

⁸⁰ Johnson-Laird ha proposto un modello teorico della mente, come macchina costruttrice di modelli, molto vicino alle idee di Craik. Secondo la teoria dei modelli mentali, la mente costruisce modelli del mondo sulla base dei quali vengono compiute le fondamentali operazioni di ragionamento; si vedano Johnson-Laird (1983), (2001), (2001).

⁸¹ Craik (1943), (1966).

⁸² Minsky (1968), MIT *Artificial Intelligence Laboratory*.

⁸³ *Artificial Intelligence. A Philosophical Introduction* (1993).

⁸⁴ John McCarthy (Boston 1927 - Stanford 2011), informatico e scienziato cognitivo statunitense, è stato l'inventore del termine 'intelligenza artificiale' (*Artificial Intelligence*) nel 1955. Inventò il linguaggio di programmazione LISP e lavorò fino al 1962 al MIT; si trasferì poi alla Stanford University dove creò lo *Stanford Artificial Intelligence Laboratory*.

⁸⁵ *The Dartmouth Summer Research Project on Artificial Intelligence* (Copeland 1993, 8).

AI, lo scopo di questa nuova disciplina era quello di "far fare alle macchine delle cose che richiederebbero l'intelligenza se fossero fatte dagli uomini".

Possiamo differenziare i due tipi di intelligenza, meccanica o artificiale (*Artificial Intelligence* - AI) vs. umana o naturale (*Natural Intelligence* - NI) ricorrendo ad alcune parole chiave:

INTELLIGENZA ARTIFICIALE - AI

computazione / algoritmo / calcolo / ricorsività / monotonicità / deducibilità / modelli

INTELLIGENZA NATURALE - NI

ragionamento / decisione / strategie / non monotonicità / fallacie / modelli parziali

Si vede immediatamente che si tratta di due dimensioni fondamentalmente diverse e non riducibili: la prima chiama in causa processi esatti di calcolo, operazioni computabili e ricorsive, capacità inferenziali di tipo deduttivo; aggettivi distintivi sono precisione, effettività, rappresentatività completa, determinismo. La seconda si situa in una sfera di competenza differente, caratterizzata dalla incertezza delle scelte, la possibilità di errore, la parzialità dell'informazione e delle rappresentazioni⁸⁶.

2.1 SISTEMI ESPERTI

Con il termine sistema esperto (*expert system*) intendiamo un programma (o un insieme di programmi) in grado di risolvere problemi e di fornire informazioni e indicazioni nell'ambito di una particolare area conoscitiva: ad esempio, una diagnosi medica, oppure la realizzazione di certe innovazioni nel campo dei mezzi di trasporto o delle telecomunicazioni. I sistemi esperti rappresentano il cuore delle ricerche e delle applicazioni della AI con centri di progettazione tecnologica diffusi in tutto il mondo. La realizzazione di questi programmi ha dato origine anche ad una nuova professione, la 'knowledge engineering' (ingegneria della conoscenza).

Una particolare categoria di sistemi esperti, sono i *sistemi intelligenti* che *imparano* come agire in un determinato ambito conoscitivo o di esperienza, in modo da potere raggiungere i propri obiettivi. In molti casi le conoscenze possedute dal sistema sono organizzate mediante regole di inferenza della forma: *se* questo, *allora* quello; *se* quello, *allora* qualcos'altro, ecc.

⁸⁶ L'incertezza nelle scelte decisionali e altri 'limiti' della razionalità umana sono stati studiati a fondo dalla psicologia; si vedano in proposito Girotto (1994), Girotto, Sperber et al. (2001), Kahneman (2000), Kahneman e Frederick (2002), Kahneman, Slovic e Tversky (1982).

Ecco due esempi di inferenze ricavate dalla base di conoscenza del sistema Mycin⁸⁷, specializzato nelle infezioni del sangue:

REGOLA DIAGNOSTICA

SE

1. il luogo della coltura è: sangue, e
2. si rilevano batteri Gram-negativi, e
3. la morfologia dell'organismo è a flagelli, e
4. il paziente presenta serie ustioni

ALLORA si ha evidenza che l'identità dell'organismo è *Pseudomonas*.

REGOLA TERAPEUTICA

SE

1. L'identità dell'organismo è *Pseudomonas*

ALLORA la terapia deve essere selezionata tra i seguenti farmaci:

1. Colistin (.98)
2. Polymyxin (.96)
3. Gentamicina (.96)
4. ecc.

Tipicamente i sistemi esperti sono caratterizzati da due tipi di proprietà:

- PROPRIETÀ MATERIALI: il supporto fisico, il veicolo attraverso cui si esplicano le funzioni del sistema
- PROPRIETÀ FORMALI: le informazioni, le funzioni e i processi elaborati dal sistema

In quest'ambito di ricerca svolgono un ruolo centrale le ontologie, che rappresentano le basi di conoscenza cui fare riferimento e che sono ottenute raccogliendo, organizzando e formalizzando enormi quantità di dati raccolti nelle osservazioni degli specifici settori. Ad esempio, un noto sistema esperto, Dendral⁸⁸, specializzato nell'analisi chimica, è stato descritto come dotato di una capacità di giudizio e ragionamento in ambito chimico al livello di quante ne potrebbe avere un laureato.

⁸⁷ Sistema esperto, introdotto nel primo periodo della AI, costruito per individuare i batteri che causano gravi infezioni, come la meningite, e indicare gli antibiotici appropriati. Si veda Copeland (1993), pp. 30-32.

⁸⁸ Progetto pionieristico nella AI degli anni '60, per la realizzazione di un sistema esperto che studiasse la formazione di ipotesi nell'ambito della scoperta scientifica; come area specifica fu scelta la chimica. Si veda Copeland (1993), p. 30.

Per entrare pienamente nel merito della caratterizzazione del concetto di *computazione*, che rappresenta il tema centrale di questa rassegna, possiamo a questo punto introdurre la seguente:

DEFINIZIONE DI COMPUTER

Macchina che elabora *simboli*, attraverso *programmi*, al fine di risolvere *problemi*.

La definizione fa ricorso ad alcuni concetti chiave, indicati in corsivo, la cui interpretazione e comprensione propone le seguenti domande, le cui risposte non sono univoche e rimandano ad approcci teorici diversi⁸⁹:

- (i) quali simboli;
- (ii) quali elaborazioni, procedure, programmi;
- (iii) quali problemi;
- (iv) quali soluzioni.

3. MACCHINE E AUTOMI

Per potere comprendere adeguatamente l'idea di computazione è necessario fare riferimento ad alcuni importanti concetti che ne rappresentano l'esplicazione: il concetto di *strategia*, il concetto di *macchina* e quello di *automa*. In termini generali, con *strategia* intendiamo la determinazione delle istruzioni (mosse, comportamenti, scelte, ecc.) che si richiedono per raggiungere un determinato obiettivo, per risolvere un dato problema.

Nell'ambito dello sviluppo tecnologico, *macchina* designa un insieme di componenti, collegati tra loro, dotati di appositi meccanismi, il cui scopo è la realizzazione di una applicazione ben determinata, compiendo un lavoro con una forza diversa da quella dell'uomo, e in genere superiore. Le tecnologie basate sull'automazione usano sistemi di controllo (circuiti logici, dispositivi elettronici, ecc.) per realizzare e gestire macchine per l'esecuzione di operazioni ripetitive e di vari gradi di complessità.

Automa è una macchina che opera in modo autonomo, come indica l'etimologia che deriva dal greco *automatos*, "che agisce di propria volontà". A questo tipo di agire si associano le caratteristiche di ripetitività e meccanicità che sostengono il diffuso uso del termine per indicare un robot.

In informatica, *automa* è un dispositivo, sotto forma di macchina *sequenziale*, creato per eseguire un particolare compito, che può trovarsi in diverse

⁸⁹ Per approfondimenti su questo argomento si vedano: Bechtel e Graham (1998), Copeland (1993), Marraffa (2002), (2003), Marraffa e Meini (2005), Minsky (1985),

configurazioni caratterizzate da una variabile che rimanda ad un determinato insieme di *stati*, e che si evolve in base alle istruzioni ricevute in ingresso, rappresentate con simboli appartenenti ad un determinato alfabeto.

Di questi concetti può essere data una caratterizzazione formale che ne evidenzia le proprietà logico-matematiche⁹⁰.

3.1 STRATEGIE

Una strategia è una serie di istruzioni per raggiungere un obiettivo. Nelle istruzioni si impone l'esecuzione di atti concreti ed elementari, per raggiungere l'obiettivo prefissato.

Le prescrizioni, o istruzioni, contenute in una strategia fanno riferimento a uno o più soggetti capaci di eseguire gli atti concreti ed elementari che esse impongono di eseguire. In particolare, si possono definire

- strategie per calcolare una operazione, dette *algoritmi* o *programmi*;
- strategie per stabilire una proprietà o una relazione, dette *test*

Una strategia per il calcolo di una operazione f (un algoritmo per f) ha l'obiettivo di trovare in un numero finito di passi, dato un qualunque oggetto a appartenente al dominio dell'operazione, il valore $f(a)$, qualora questo valore esista.

Una strategia per il calcolo di una operazione n -aria f (un algoritmo per f) ha l'obiettivo di trovare in un numero finito di passi, data una qualunque *ennumpla* ordinata di oggetti a_1, \dots, a_n appartenente al dominio dell'operazione, il valore $f(a_1, \dots, a_n)$ qualora questo valore esista.

Una strategia per stabilire una proprietà P (un *test* per P) ha l'obiettivo di trovare in un numero finito di passi, dato un qualunque oggetto a appartenente al dominio della proprietà P , una dimostrazione o una refutazione della proposizione

$P(a) = a$ ha la proprietà P

Una strategia che serve a stabilire una relazione R (un test per R) ha l'obiettivo di trovare, in un numero finito di passi, data una qualunque *ennumpla* ordinata di oggetti a_1, \dots, a_n , appartenente al dominio della relazione R , una

⁹⁰ Per approfondimenti si rimanda ad Abrusci (2009); per i concetti di *dimostrazione* e *refutazione*, già affrontati nel primo capitolo di questo volume, si vedano anche Dalla Chiara et al (2004), Frixione e Palladino (2004), Moriconi (2009).

dimostrazione o una refutazione della proposizione $R(\mathbf{a}_1, \dots, \mathbf{a}_n)$, vale a dire, gli oggetti $\mathbf{a}_1, \dots, \mathbf{a}_n$ stanno in questo ordine nella relazione R .

Le strategie sono *concrete* e *finite* quando consistono in atti concreti eseguiti in un numero finito di passi. Le strategie possono essere *meccanizzate* (cioè affidate ad una macchina che sappia compiere gli atti concreti previsti nella strategia), e chi esegue una strategia, che sia una macchina o una persona, comunque agisce in modo meccanico.

3.2 MACCHINE

Il concetto generale di macchina è strettamente collegato a quello di strategia: una macchina è un qualunque “soggetto” che sa eseguire una strategia o che interviene nella esecuzione di una strategia. Per ciascuna strategia si può costruire una macchina che la esegue: è sufficiente che la macchina sappia compiere tutti gli atti concreti ed elementari che sono previsti nelle istruzioni che costituiscono la strategia⁹¹.

Una strategia non è tale se non è eseguibile da una macchina o da più macchine in collaborazione tra loro. Ad esempio, le macchine calcolatrici sono macchine che sanno eseguire molte strategie (operazioni aritmetiche), altre macchine invece sanno eseguire soltanto una strategia.

“Una strategia per il gioco degli scacchi è sicuramente frutto di grande intelligenza, ma una volta trovata la si può affidare anche a uno del tutto inesperto di scacchi il quale è tenuto soltanto a osservarla *così come è*, a eseguirla senza metterci nulla di creazione personale (anzi, se volesse modificarla, lui inesperto, la strategia non funzionerebbe più così bene...), a eseguirla *come se fosse una macchina*: tanto vale allora che la strategia venga affidata ad una macchina. Una macchina creata per un uno scopo non fa altro che eseguire una strategia (pensata dal suo costruttore).”⁹²

Le nozioni di macchina e di strategia sono concetti teorici essenziali nell’ambito dell’informatica e dell’intelligenza artificiale e ricoprono un ruolo fondamentale per comprendere correttamente l’idea di *computazione* introdotta negli anni ‘50 da Alan Turing⁹³.

⁹¹ Abrusci (2009), pp. 42-46.

⁹² Ibid. p. 44.

⁹³ Alan Mathison Turing (Londra 1912 - Wilmslow 1954), logico e matematico britannico, ha contribuito in modo essenziale con la sua opera alla nascita dell’informatica, alle ricerche sulla cibernetica e l’intelligenza artificiale. Durante la Seconda Guerra Mondiale si impegnò con successo, come membro del gruppo di crittografi stabilitosi a Bletchley Park, nella soluzione dei problemi di decifrazione dei codici nemici. Con il suo lavoro ha contribuito in modo

3.3 MACCHINA DI TURING E COMPUTABILITÀ

We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity, like the playing of chess, would be best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried⁹⁴.

3.3.1 *Turing e l'idea di Intelligenza Artificiale*

Ad Alan Turing si devono le intuizioni fondamentali che portarono allo sviluppo dell'intelligenza artificiale, molti anni prima della effettiva nascita di questa disciplina⁹⁵. Nel celebre articolo pubblicato su *Mind*, Turing si chiede se le macchine possano pensare: “I propose to consider the question ‘Can machines think?’” e dopo avere sottoposto il problema ad una attenta analisi filosofica, arriva alla conclusione che alla fine del XX secolo l'opinione generale sarà così mutata che si parlerà senza problemi di macchine pensanti, senza aspettarsi di essere contraddetti⁹⁶.

Quando Turing scrisse questo articolo, nel mondo erano stati realizzati pochi elaboratori elettronici⁹⁷, che già venivano chiamati “cervelli elettronici”, e cominciava ad affermarsi l'idea che da questi prototipi si sarebbero sviluppate delle vere e proprie macchine pensanti. Tuttavia egli non anticipò solamente di parecchi anni la nascita dell'intelligenza artificiale; il suo contributo più significativo è rappresentato dall'aver concepito sul piano teorico l'idea esatta di computer, scrivendo nel 1936-37 un lavoro sui fondamenti logici della computazione⁹⁸ nel quale viene impiegato il modello di macchina astratta o *macchina universale* oggi conosciuta come “macchina di Turing”⁹⁹.

determinante alla sviluppo della teoria della computabilità e alla progettazione delle macchine elettroniche programmabili realizzate nella seconda metà del XX secolo.

⁹⁴ Turing, *Computing Machinery and Intelligence* (1950, p. 460).

⁹⁵ Per approfondimenti si vedano Copeland (1993), Numerico (2005).

⁹⁶ Turing (1950), p. 460; trad. it. in Somenzi e Cordeschi (1986), p. 193.

⁹⁷ Il Mark I di Manchester, l'EDSAC di Cambridge in Gran Bretagna, l'ENIAC e il BINAC negli Stati Uniti.

⁹⁸ *On computable numbers, with an application to the Entscheidungsproblem* (Turing 1937).

⁹⁹ Tale macchina, presentata in Turing (1936), ha consentito di dare una risposta negativa al problema della decidibilità della logica del primo ordine, l'“Entscheidungsproblem”, posto da David Hilbert nel 1928. Tale risposta negativa (teorema di Church-Turing) venne data sia da Alonzo Church nel 1936, che da Turing, indipendentemente nel 1937, facendo ricorso alle tecniche utilizzate da Kurt Gödel per dimostrare il teorema di incompletezza dell'aritmetica.

3.3.2 *Macchina di Turing*

Una macchina di Turing è una macchina logica e ideale, che ha ispirato la costruzione di macchine concrete¹⁰⁰; sa eseguire da sola - senza interazioni con altri agenti - una sola strategia, più precisamente un solo algoritmo per il calcolo di una funzione sui numeri naturali. Una macchina di Turing è capace di eseguire - via la codificazione - anche strategie per il calcolo di funzioni su insiemi diversi da quello dei numeri naturali.

Una seria limitazione è data dal fatto che una macchina di Turing non prevede interazione o collaborazione con altri soggetti. Il fatto che ciascuna macchina di Turing sa eseguire un solo programma, un solo algoritmo, la differenzia dalle usuali macchine calcolatrici (che possono svolgere vari programmi, anche se uno solo alla volta) e soprattutto dagli uomini che possono eseguire vari tipi di calcolo.

Turing affrontò il problema di fornire un equivalente rigoroso del concetto intuitivo di *algoritmo* definendo un modello dell'attività di un essere umano che stia eseguendo un calcolo. Egli elaborò tale modello nella forma di una classe di dispositivi computazionali, macchine calcolatrici astratte, che in seguito furono chiamate *macchine di Turing* (MT).

Le MT sono macchine astratte nel senso che, nel caratterizzarle, non vengono presi in considerazione quei vincoli che sono fondamentali in una macchina calcolatrice reale (le dimensioni della memoria, i tempi del calcolo, ecc.), e sono definite a prescindere dalla loro realizzazione fisica.

Una MT agisce sopra un *nastro*: una sequenza di caselle nelle quali possono essere registrati simboli di un determinato *alfabeto finito*; essa è dotata di una *testina* (I/O) con cui è in grado di effettuare operazioni di lettura e scrittura su una casella del nastro. La macchina si evolve nel tempo e ad ogni istante si può trovare in un certo *stato interno*, parte di un insieme finito di stati. Inizialmente sul nastro viene posta una *stringa* che rappresenta i dati che caratterizzano il problema che viene sottoposto alla macchina.

¹⁰⁰ Partendo dal modello della macchina di Turing, Max Newman progettò una macchina chiamata *Colossus* (primo computer elettronico programmabile, sebbene limitato ad un solo scopo - *special-purpose*) che decifrava in modo veloce ed efficiente i codici creati con i modelli di macchine cifratrici tedesche, che perfezionavano la celebre macchina *Enigma*. Alla fine della guerra il gruppo di Bletcheley si sciolse e la ricerca britannica sull'automazione si trasferì presso l'università di Manchester dove venne creato il modello di computer *general purpose* Mark I, fatto funzionare nel 1948. Negli Stati Uniti, la storia dell'automazione inizia con una macchina chiamata ENIAC (*Electronic Numerical Integrator And Computer*) costruita alla Pennsylvania University da J. Mauchly e J. P. Eckert nel 1945.

La macchina è dotata anche di un repertorio finito di istruzioni che determinano la sua evoluzione in conseguenza dei dati iniziali. L'evoluzione si sviluppa per passi successivi che corrispondono a una sequenza discreta di istanti successivi. Le proprietà precedenti sono comuni a molte macchine formali (automi a stati finiti, automi a pila, ecc.). Caratteristica di una MT è di disporre di un nastro potenzialmente infinito, cioè estendibile quanto si vuole, qualora questo si renda necessario.

La MT utilizza per la “scrittura” un *nastro* monodimensionale di lunghezza virtualmente illimitata in entrambe le direzioni (tuttavia, in ogni fase del calcolo, la macchina potrà disporre soltanto di una porzione finita di tale nastro). Il nastro è suddiviso in *celle*, ciascuna delle quali potrà ospitare un solo *simbolo* alla volta. La macchina potrà disporre soltanto di un insieme finito di simboli, chiamato l'*alfabeto* di quella macchina.

Il fatto che l'alfabeto sia finito non rappresenta una limitazione: è sempre possibile rappresentare un nuovo simbolo mediante una sequenza finita di simboli dell'alfabeto, ed avere così la possibilità di esprimere un numero virtualmente *infinito* di simboli:

sia $\Sigma = \{s_1, s_2, \dots, s_n\}$ l'alfabeto di una MT ,

ogni cella del nastro potrà contenere uno di tali simboli, oppure, in alternativa, restare vuota (si indica con s_0 la cella vuota).

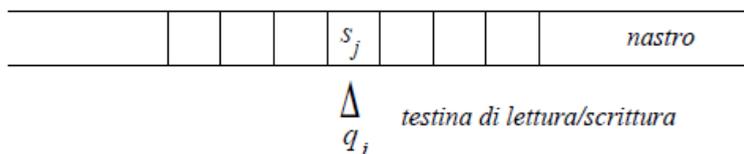


FIGURA 15: Nastro di una MT ad un certo stato

La macchina può esaminare soltanto una cella alla volta, ed “osservare” ad ogni passo un singolo simbolo. A tal fine la macchina è dotata di una testina di lettura, che si colloca, in ogni fase del calcolo, su di una singola cella e può spostarsi verso destra o verso sinistra per accedere ad altre celle.

Si possono scrivere nuovi simboli, cancellare quelli già scritti o sostituirli con altri. La testina eseguirà anche tali compiti di cancellazione e di scrittura, agendo ogni volta su una singola cella.

Siano q_0, q_1, \dots, q_m gli stati che una MT può assumere,

nella rappresentazione del nastro, sotto la testina di lettura/scrittura viene indicato lo stato della macchina nella fase di calcolo presa in considerazione.

Si definisce *configurazione* di una MT, in un dato stadio del calcolo, la coppia costituita dallo stato *interno* che essa ha in quel momento e dal simbolo osservato dalla testina: ad esempio, la configurazione della macchina nella FIGURA 15 è (q_i, s_j) .

Una MT può eseguire le seguenti operazioni, consistenti in spostamenti della testina lungo il nastro, scrittura e cancellazione di simboli, mutamenti dello stato interno:

- 1) sostituzione del simbolo osservato con un altro simbolo (eventualmente con il simbolo s_0 ; in tal caso si ha la cancellazione del simbolo osservato);
- 2) spostamento della testina su di una delle celle immediatamente attigue del nastro;
- 3) cambiamento dello stato interno della macchina.

Possiamo rappresentare ogni singola operazione della macchina (operazione atomica) mediante una *terna*, formata da: (i) il simbolo che deve essere scritto sulla cella osservata, (ii) quale spostamento deve essere eseguito (S, D, C)¹⁰¹, (iii) lo stato che la macchina deve assumere alla fine dell'operazione.

Ad esempio, la terna: $(s_i \text{ S } q_j)$,

significa che la macchina deve scrivere il simbolo s_i sulla cella osservata, spostarsi di una cella a sinistra, ed assumere infine lo stato q_j .

Con la macchina di Turing (che riassume la struttura funzionale di un computer) è possibile risolvere tanto problemi numerici che problemi non numerici, associando ai simboli un significato alfabetico. Le operazioni che possono essere calcolate da una macchina di Turing sono ovviamente calcolabili da qualsiasi macchina, con qualche programma. Si dimostra che il concetto di calcolabilità secondo Turing esprime pienamente il concetto intuitivo di calcolabilità, e può a tutti gli effetti essere usato al suo posto. Questo è il contenuto della proposizione nota come tesi di Church-Turing¹⁰²,

¹⁰¹ Rispettivamente, *Sinistra, Destra, Centro*.

¹⁰² Nota anche come tesi di Church-Turing, consiste nella affermazione che “se un problema è intuitivamente calcolabile, allora esisterà una macchina di Turing (o un dispositivo equivalente) in grado di risolverlo (cioè di calcolarlo)”. Formalmente, equivale a dire che la classe delle funzioni calcolabili coincide con quella delle funzioni calcolabili da una macchina di Turing.

accettata da tutta la comunità scientifica, che asserisce che vale anche l'inverso: ogni funzione che sia calcolabile (da qualche macchina, con qualche programma) è anche calcolabile da una Macchina di Turing.

Il concetto di macchina di Turing ha ispirato John von Neumann¹⁰³ a elaborare il modello, chiamato *Macchina di von Neumann*, che dal 1946 è stato preso come punto di riferimento per la costruzione e il funzionamento delle macchine calcolatrici e dei calcolatori elettronici. Pertanto, ciascuna concreta macchina calcolatrice o calcolatore elettronico è una macchina di von Neumann, ossia è costruita in analogia con quel modello, a sua volta ispirato dal concetto di macchina di Turing.

Una macchina di von Neumann ha le seguenti caratteristiche generali:

- 1) è un processore *sequenziale*: esegue le istruzioni del suo programma una dopo l'altra;
- 2) ciascuna stringa di simboli è depositata in una specifica area di memoria;
- 3) l'accesso a una stringa depositata avviene sempre attraverso l'indirizzo numerico della locazione della stringa;
- 4) Esiste un unico centro di controllo, la *Central processing Unit* o CPU.

Va tenuto presente che l'architettura di von Neumann è solo un particolare approccio alla costruzione di sistemi simbolici universali, che si è ampiamente sviluppato nel corso del XX secolo, ma che non rappresenta l'unico approccio possibile. Infatti, non vi è una connessione necessaria tra la manipolazione di simboli e il fatto che il processo sia *sequenziale*. Negli ultimi decenni si sono avuti notevoli progressi nella progettazione e realizzazione di sistemi che operano in *parallelo*, in cui il controllo centralizzato viene sostituito da sistemi di cooperazione a molti livelli o modulari¹⁰⁴.

La FIGURA 16 mostra una rappresentazione schematica della architettura di von Neumann: l'Unità di Controllo (*Control Unit*) e l'unità Logico-aritmetica (*Arithmetic-Logic Unit*) formano i componenti principali della CPU (*Central Processing Unit*).

¹⁰³ John von Neumann partecipò alla realizzazione del progetto ENIAC di Eckert e Mauchly, e divenne presto l'esponente più noto della nuova tecnologia dell'automazione attraverso l'enunciazione dei principi generali di organizzazione, cui aderirono tutte le generazioni successive di macchine; per questa ragione i moderni computer standard sono noti come macchine di von Neumann.

¹⁰⁴ Si vedano Copeland (1993), Abrusci (2009) Frixione e Palladino (2004).

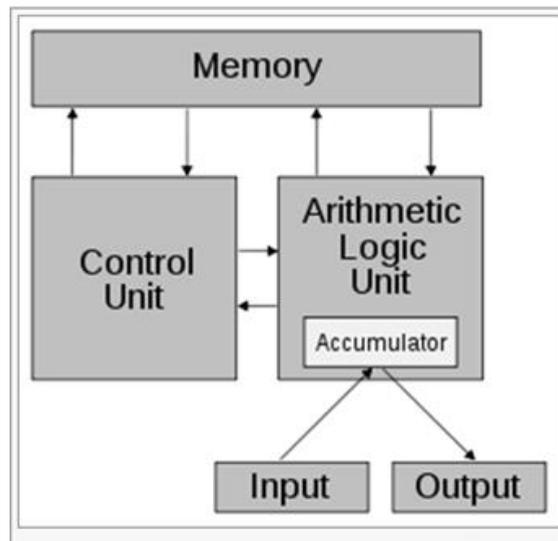


FIGURA 16: Architettura di von Neumann

E' noto che ci sono, e lo si può dimostrare, funzioni non calcolabili, ossia che mai nessuna macchina potrà calcolare: non tutte le operazioni possono essere calcolate in maniera meccanica, con un programma, da una macchina. Lo studio della calcolabilità o della complessità computazionale si collega all'analisi del rapporto tra *determinismo* e *non determinismo*, con la distinzione tra ciò che è fattibile e ciò che non lo è (anche se è calcolabile)¹⁰⁵.

4. ESPERIMENTI MENTALI

Un esperimento mentale (*Gedankenexperiment*)¹⁰⁶, è un esperimento disegnato teoricamente e che non si intende realizzare praticamente, ma viene solo immaginato: i suoi risultati non vengono quindi misurati, ma calcolati teoricamente in base alle leggi della matematica o della fisica. In questa sezione presentiamo due noti esperimenti mentali, creati rispettivamente da Alan Turing e dal filosofo John Searle, dedicati al problema dell'*intelligenza* delle macchine.

¹⁰⁵ Per approfondimenti si rimanda ad Abrusci (2009), Barker-Plummer (2011).

¹⁰⁶ Il termine si deve al fisico danese Hans Christian Ørsted (1777 - 1851). Un noto esperimento mentale è quello del treno, proposto da Einstein per spiegare la teoria della relatività. Si veda Buzzoni (2204).

4.1 TEST DI TURING

Nell'articolo *Computing Machinery and Intelligence* del 1950, Turing propone un “esperimento mentale” di confronto tra una mente “umana” e una mente “artificiale”, conosciuto come *Test di Turing*.

Facendo riferimento a calcolatori reali, ma caratterizzati in maniera analoga a macchine di Turing, Turing si dichiara fiducioso che macchine di questo tipo possano giungere a simulare, nel volgere di pochi decenni, non soltanto il “comportamento computazionale” di un essere umano, ma anche qualsiasi altra attività cognitiva umana.

Turing si propone di rispondere alla domanda “possono pensare le macchine?” proponendo il *gioco dell'imitazione*.

Il gioco viene giocato da tre “attori” o “agenti”:

- a) un essere umano,
- b) una macchina calcolatrice / calcolatore elettronico
- c) un altro essere umano, l'interrogante.

L'interrogante non può vedere a) e b), non sa chi dei due sia l'essere umano, e può comunicare con loro solo in maniera indiretta, p. es. attraverso un terminale video e una tastiera posti al di là di uno schermo.

L'interrogante deve presentare agli interlocutori (a uomo) e (b macchina) delle domande, in maniera tale da scoprire nel più breve tempo possibile quale dei due sia l'uomo e quale la macchina. Nello svolgimento dell'esperimento

a) si comporterà in modo da agevolare c), mentre b) dovrà rispondere in modo da ingannare c) il più a lungo possibile.

Invece di chiedersi se le macchine possono pensare, dice Turing, è più corretto chiedersi se una macchina possa battere un uomo nel gioco dell'imitazione, o, comunque, quanto a lungo possa resistergli. A suo parere, nell'arco di 50 anni i calcolatori elettronici avrebbero raggiunto un livello tale di perfezionamento da essere in grado di trarre in inganno l'interrogante umano con probabilità sempre maggiore.¹⁰⁷

¹⁰⁷ Per approfondimenti si vedano Copeland (1993), Numerico (2005), Hodges (2007), Oppy e Dowe (2011).

4.2 LA STANZA CINESE DI SEARLE

L'esperimento mentale detto della *stanza cinese* è stato ideato da John Searle¹⁰⁸ in risposta al problema sollevato dal test di Turing. Viene considerato un contro-esempio della teoria dell'intelligenza artificiale forte, in base a cui le macchine possono avere una intelligenza propria e autonoma dal creatore umano. Alla base del disegno sperimentale proposto da Searle vi è la distinzione tra la sintassi, come organizzazione della forma grammaticale, e la semantica, come organizzazione del significato, espressione della cognizione.

Searle presentò l'argomentazione della Stanza cinese nel suo articolo *Minds, Brains and Programs*¹⁰⁹. Da allora, è stato uno dei temi forti del dibattito sull'ipotesi teorica, chiamata dallo stesso Searle "intelligenza artificiale forte": *According to strong AI, the computer is not merely a tool in the study of the mind; rather, the appropriately programmed computer really is a mind.*¹¹⁰

Secondo i sostenitori dell'intelligenza artificiale forte un computer, adeguatamente programmato, non è soltanto una simulazione o un modello della mente, ma può essere una *mente* a tutti gli effetti: capisce, ha condizioni conoscitive e può pensare e risolvere problemi. L'esperimento mentale concepito da Searle si oppone a questa posizione.

L'argomentazione della stanza cinese è la seguente:

Si supponga che si possa costruire un computer che si comporti come se capisse la lingua cinese: il computer prende dei caratteri cinesi in ingresso, e produce altri caratteri cinesi in uscita, dopo avere consultato una appropriata tabella.

Si supponga che il comportamento di questo computer sia così convincente da poter facilmente superare il test di Turing: il computer può convincere un essere umano che parla correttamente la lingua cinese (per esempio un cinese) di parlare con un altro essere umano che parla correttamente cinese, mentre in realtà sta parlando con un calcolatore. A tutte le domande dell'essere umano il computer risponderebbe appropriatamente, in modo tale

¹⁰⁸ John R. Searle (1932 Denver, Colorado), filosofo statunitense, insegna alla University of California, Berkeley; ha dato contributi fondamentali alla filosofia del linguaggio, in particolare nell'ambito della teoria degli atti linguistici, alla filosofia della mente e alle scienze sociali.

¹⁰⁹ *Menti, cervelli e programmi*, pubblicato nel 1980 dalla rivista scientifica *Behavioral and Brain Sciences* (e in lingua italiana da Le Scienze).

¹¹⁰ Searle (1980).

che questa persona si convinca di parlare con un altro essere umano che parla correttamente il cinese.

I sostenitori dell'intelligenza artificiale forte concluderebbero che il computer *capisce* la lingua cinese, come farebbe un essere umano, in quanto non c'è nessuna differenza tra il comportamento della macchina e di un uomo o donna che conosce la lingua cinese.

A questo punto, Searle chiede di supporre che lui (un uomo) venga messo al posto della macchina: si siede all'interno del calcolatore. Immagina di essere in una piccola stanza (la stanza cinese) dalla quale riceve dei simboli cinesi, e una tabella che gli consente di produrre dei simboli cinesi in uscita in modo identico a quanto faceva il programma seguito dal calcolatore elettronico.

Searle fa notare che egli non capisce i simboli cinesi, segue solo le istruzioni della tabella. Quindi la sua mancanza di comprensione dimostra che il calcolatore non può comprendere il cinese, poiché la macchina si trova nella sua stessa situazione.

Il computer è un *semplice manipolatore di simboli*, esattamente come lo è l'uomo nella stanza cinese – per questa ragione il computer non è in grado di capire le informazioni che si stanno trasmettendo tanto quanto non lo è l'essere umano.

Più in generale, Searle mette in evidenza la differenza tra le operazioni di manipolazione di simboli, che vanno sotto la definizione di *sintassi*, e le operazioni e funzioni cognitive di organizzazione e comprensione dei significati, designate col termine *semantica*, che sono proprietà distintive del linguaggio umano¹¹¹.

5. AUTOMI

Come abbiamo detto, in informatica con automa si intende una macchina sequenziale, creata per eseguire un particolare compito, che può trovarsi in diverse configurazioni, più o meno complesse, caratterizzate da un determinato insieme di stati, ed evolve in base alle istruzioni ricevute in

¹¹¹ Per approfondimenti si vedano Searle (1980), (1998), Fodor (2001), Copeland (1993), Pessa e Penna (2000), Marraffa (2002), Cummins (1989).

ingresso (stato iniziale), sulla base di simboli appartenenti ad un determinato alfabeto.

Quando l'automa si trova in un dato stato, esso può accettare solo un sottoinsieme dei simboli del suo alfabeto. L'evoluzione di un automa parte da un particolare stato, detto stato *iniziale*. Un sottoinsieme privilegiato dei suoi stati è detto insieme degli stati finali.

Un automa si definisce anche come un sistema *dinamico* (un sistema che *evolve nel tempo*, in cui sia l'ingresso che l'uscita si sviluppano nel tempo); oppure *discreto* (nella scansione del tempo e nella descrizione del suo stato) e *invariante* (il sistema si comporta sempre alla stessa maniera). Un automa è *deterministico* se, dato uno stato iniziale ed un simbolo in ingresso, è possibile una sola transizione.

5.1 AUTOMI E LINGUAGGI

Gli automi sono spesso utilizzati per descrivere linguaggi formali, agendo come accettori o riconoscitori di un linguaggio.

Con linguaggio formale si intende un insieme di *stringhe* di lunghezza finita, costruite in base ad un *alfabeto* finito, che consiste in un insieme finito *caratteri, simboli* o *lettere*. Di un linguaggio formale può fare parte la stringa o parola vuota, cioè la sequenza costituita da zero caratteri: questa viene denotata come ϵ , ε o λ . Un linguaggio può essere finito o infinito in quanto non si pongono limiti alla lunghezza delle stringhe. Un tipico alfabeto potrebbe essere $A := \{a, b\}$ e tipiche stringhe su questo alfabeto:

ababba aaabbbaaa aaabbbabaababb

Una sequenza di simboli (detto anche stringa o parola) appartiene al linguaggio se essa viene accettata dal corrispondente automa, cioè se porta l'automa in uno stato valido, che sia lo stesso o un altro stato. Un sottoinsieme del linguaggio riconosciuto, chiamato *linguaggio marcato* porta l'automa dal suo stato iniziale ad uno stato finale o marcato. A diverse classi di automi corrispondono diverse classi di linguaggi, caratterizzate da diversi livelli di complessità.

I seguenti sono alcuni tipi di automi che sono di particolare interesse e impiego nell'ambito degli studi sull'intelligenza artificiale e l'analisi automatica dei linguaggi naturali (*natural language processing*).¹¹²

5.1.1 Automi a Stati Finiti (*Finite State Automata*)

Sono dotati di un insieme finito di stati, leggono una stringa di simboli in ingresso (simbolo per simbolo) in maniera ordinata per decidere se essa appartenga o meno ad un certo linguaggio.

Formalmente tali automi sono delle quintuple, (Q, Σ, f, q_0, F) , formate da un *alfabeto* finito dei simboli in ingresso (Σ), un insieme finito di *stati* (Q) tra cui si distingue uno *stato iniziale* (q_0) ed un sottoinsieme di stati, detti *finali* (F), ed una *funzione di transizione* (f). Tale funzione, descritta mediante una tabella di transizione degli stati, è definita per coppie (stato corrente, simbolo scandito) e stabilisce la transizione da compiere, ossia lo stato in cui si transita leggendo il dato simbolo.

Il funzionamento dell'automa può essere così descritto: partendo dallo stato iniziale e dal primo simbolo della stringa in ingresso si decide, in base alla funzione assegnata, di transitare in un determinato stato (potrebbe anche essere lo stesso stato). Finché esiste un altro simbolo nella stringa da scandire si opera alla stessa maniera fino ad esaurire la stringa in ingresso. La stringa si dirà *accettata* se si giunge in uno stato appartenente al sottoinsieme degli stati finali. Tali automi sono in grado di riconoscere i linguaggi regolari (particolari tipi di linguaggi formali).

5.1.2 Automi a Pila (*Push Down Automata*)

Gli automi possono anche essere dotati di memoria supplementare (rispetto ai soli stati) che utilizza una struttura dati che ha la forma di una pila (*stack*); i dati vengono estratti (letti) in ordine rigorosamente inverso rispetto a quello in cui sono stati inseriti (scritti), in analogia con il modo in cui viene formata una pila di piatti.

Tali automi sono in grado di riconoscere una classe più ampia di linguaggi rispetto agli automi a stati finiti, come quella dei linguaggi liberi dal contesto (*Context Free Languages*)¹¹³.

¹¹² Per approfondimenti si vedano Hopcroft e Ullman (1979), Croft e Cruse (2010).

¹¹³ Sono i linguaggi generati da una grammatica *Context Free* (CF): una grammatica formale in cui ogni regola sintattica (*production rule*) ha la forma: $V \rightarrow w$, dove V è un simbolo non-terminale e w è una stringa di simboli terminali e/o non-terminali.

Lo stato degli automi a pila è costituita da una pila di simboli. Solo il simbolo in cima alla pila in un dato momento è accessibile e può essere letto. Le transizioni negli automi a pila dipendono dal simbolo in ingresso e dal simbolo in cima alla pila; una transizione può comportare il deposito di un nuovo simbolo in cima alla pila (operazione *push*) e/o l'emissione di un simbolo in uscita (operazione *pop*). L'insieme di questi automi include l'insieme degli automi a stati finiti.

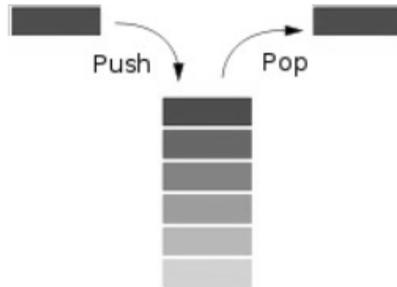


FIGURA 17: *Automa a pila*

5.1.3 *Macchine di Turing*

Rappresentano il massimo livello di complessità raggiungibile da un automa, e generalizzano sia gli automi a pila che gli automi a stati finiti.

5.1.4 *Automi non Deterministici*

Vengono studiati anche automi non deterministici: dato uno stato dell'automata ed un simbolo in ingresso è possibile più di una transizione. Tali automi vengono studiati e impiegati nella Teoria della complessità.

6. AUTOMI, LINGUAGGI E GRAMMATICHE

Lo studio delle proprietà degli automi si collega allo studio delle proprietà formali e combinatorie tanto dei linguaggi artificiali che delle lingue naturali. Particolare attenzione è dedicata al carattere *generativo* dei linguaggi (ricorsività): dato un insieme finito di *parole* (stringhe) può essere prodotto un insieme finito di *enunciati* (frasi).

Una grammatica è un dispositivo teorico che consente di specificare (con mezzi finiti) quali sono le espressioni sintatticamente corrette (grammaticali), e in particolare le frasi, di un linguaggio. Avrà un alfabeto, una definizione di stringa, e un insieme di regole per la generazione (derivazione, inferenza) dell'insieme delle stringhe grammaticali.

Una *grammatica* G per un linguaggio L è uno strumento per specificare in modo rigoroso quali sono le stringhe grammaticali di L rispetto a tutte le stringhe che si possono scrivere con lo stesso alfabeto.

6.1 GERARCHIA DI CHOMSKY

Le proprietà formali di vari tipi di grammatiche sono state studiate da Noam Chomsky nel periodo tra il 1950 e il 1960, con particolare riferimento al tipo di grammatiche *Context Free* (CF) da lui chiamate *Phrase Structure Grammars*, che possiamo tradurre come grammatiche a struttura sintagmatica¹¹⁴. Nel modello di grammatica per il linguaggio naturale formulato da Chomsky in questo periodo, noto come modello *generativo trasformativale*, la sintassi di una lingua viene descritta da un insieme di regole *context free* associate ad un insieme di regole *trasformativali*.

Come esempio, consideriamo il seguente alfabeto:

$S = \{il, un, topo, gatto, rincorre, mangia, dorme, canta\}$

Vogliamo individuare una grammatica che consenta di caratterizzare quelle stringhe sull'alfabeto S che corrispondono a frasi sintatticamente corrette dell'italiano: p. es. *il topo salta* oppure *un gatto rincorre il topo*, distinguendole dalle stringhe che non corrispondono a espressioni sintatticamente corrette dell'italiano: p. es. *un un topo canta gatto* o non costituiscono frasi complete: p. es. *il gatto*.

Una grammatica del genere può essere formulata come segue:

- | | |
|---|--------------------------------------|
| (1) $E \rightarrow SN SV$ | (7) $VI \rightarrow dorme$ |
| (2) $SN \rightarrow \text{Articolo Nome}$ | (8) $VI \rightarrow salta$ |
| (3) $SV \rightarrow VT SN$ | (9) $\text{Art} \rightarrow il$ |
| (4) $SV \rightarrow VI$ | (10) $\text{Art} \rightarrow un$ |
| (5) $VT \rightarrow rincorre$ | (11) $\text{Nome} \rightarrow gatto$ |
| (6) $VT \rightarrow mangia$ | (12) $\text{Nome} \rightarrow topo$ |

¹¹⁴ Si vedano Chomsky (1956), (1957), (1963), (1965), (1977).

La seguente gerarchia di grammatiche e corrispondenti linguaggi formali e automi è stata individuata dagli studi di Chomsky¹¹⁵:

Gerarchia di Chomsky	Grammatica	Linguaggio	Automa
Tipo – 0		Ricorsivamente enumerabile	Macchina di Turing
Tipo – 1	Context Sensitive	Context Sensitive	Automa lineare
Tipo – 2	Context Free	Context Free	Automa a pila
Tipo – 3	Regolare	Regolare	A stati finiti

FIGURA 18: *Gerarchia di Chomsky*

Le grammatiche di tipo-0, o grammatiche illimitate, includono le grammatiche formali che generano esattamente tutti i linguaggi che possono essere accettati da una Macchina di Turing, i linguaggi ricorsivamente enumerabili.

Le grammatiche di tipo-1 o grammatiche dipendenti dal contesto (*Context Sensitive*) generano linguaggi dipendenti dal contesto che hanno regole della forma: $\alpha A \beta \rightarrow \alpha \gamma \beta$, dove A è un simbolo non terminale, α , β , γ sono stringhe di simboli terminali e non terminali; le stringhe α , β possono essere vuote, ma γ non può essere vuota.

Infine, le grammatiche di tipo-2 sono quelle libere dal contesto (*Context Free*) e generano linguaggi con questa proprietà. Sono definite da regole nella forma $A \rightarrow \gamma$, dove A è un simbolo non terminale e γ è una stringa di simboli terminali e non terminali. Questi linguaggi sono esattamente quelli riconosciuti da un automa a pila non deterministico e sono alla base della costruzione di grammatiche formali per i linguaggi naturali.

¹¹⁵ Noam Chomsky (1928 Philadelphia, Pennsylvania), linguista, filosofo e scienziato cognitivo statunitense, ha contribuito in modo essenziale con le sue ricerche sulle lingue naturali alla formulazione della teoria generativo-trasformazionale della sintassi. Oltre alla gerarchia qui menzionata, si devono a lui alcuni importanti risultati, in particolare la concezione della sintassi come grammatica formale basata su alcuni principi innati (teoria innatista); si vedano Chomsky (1981), (2010). Insegna nel *Department of Linguistic and Philosophy* del Massachusetts Institute of Technology (MIT).

ESERCIZI

Gli esercizi di questa sezione sono intesi come suggerimenti per ricerche e approfondimenti

E1 Cibernetica

esercizio 1

Fornire esempi di retroazione positiva e negativa, facendo particolare riferimento alla psicologia, le scienze della vita, le neuroscienze

esercizio 2

Approfondire la relazione tra cibernetica e comunicazione, prendendo in considerazione gli autori menzionati nel testo e le relative indicazioni bibliografiche.

esercizio 3

Cercare approfondimenti sulle teorie cognitive di Norbert Wiener e Kenneth Craik in base a cui il cervello umano opererebbe in modo simile a quello delle macchine.

E2 Intelligenza Artificiale

esercizio 1

Approfondire le proprietà che contraddistinguono e differenziano Intelligenza Artificiale e Intelligenza Naturale. Cercare esempi di realizzazioni tecnologiche a partire dalla seconda metà del XX secolo.

esercizio 2

Cercare informazioni sul programma di interazione attraverso computer noto come “Eliza the Psychotherapist”, creato da J. Weizenbaum nel 1966 presso il MIT (si veda in proposito Copeland 1993).

esercizio 3

Cercare informazioni su altri programmi “intelligenti” come “Shrdlu” creato nel 1968 da T. Winograd presso l’*Artificial Intelligence Laboratory* del MIT (si veda in proposito Copeland 1993).

esercizio 4

Cercare esempi di sistemi esperti impiegati in particolare nel campo della psicologia e della medicina, o nell’area delle scienze umane.

E3 Macchine e automi

esercizio 1

Cercare esempi di strategie e di macchine che si applichino al campo delle scienze umane e delle scienze della vita.

esercizio 2

Approfondire il problema dell'automazione applicata alla valutazione in psicologia e nelle scienze sociali. E' possibile utilizzare il computer nella realizzazione dei test psicologici? Per approfondimenti di questo argomento si rimanda a Argentero (1993).

esercizio 3

Fare ricerche in merito agli sviluppi del *Natural Language Processing* e dei progressi raggiunti in questi anni rispetto alle ipotesi teoriche individuate nella seconda metà del XX secolo. Includere nell'approfondimento anche il problema della traduzione automatica (automatic translation).

esercizio 4

Approfondire la questione della formulazione di grammatiche automatizzate per il linguaggio naturale e dello sviluppo di grammatiche a base logica in questo campo; si faccia riferimento p. es. a Bar-Hillel (1959), (1960), Chomsky (1956), (1957), (1981), Lambek (1958), (1961).

BIBLIOGRAFIA

- ABRUSCI V. M. (2009), *Logica. Lezioni di primo livello*, Cedam, Milano.
- ANOLLI L., LEGRENZI P., (2001), *Psicologia generale*, Il Mulino, Bologna.
- ARGENTERO P. (1993), *Test e computer. La valutazione psicologica assistita dall'elaboratore*, Franco Angeli, Milano.
- BARKER-PLUMMER D. (2011), *Turing Machines*, The Stanford Encyclopedia of Philosophy, E. N. Zalta (ed.), URL: <<http://plato.stanford.edu/archives/spr2011>>
- BAR-HILLEL Y. (1959), *Decision Procedures for Structure in Natural Language*, "Logique et Analyse", 2, pp.19-29; trad. it. *Procedimenti di decisione per la struttura nelle lingue naturali*, in A. De Palma (a cura di)(1974), pp. 311-329.
- ID. (1960), *Some Linguistic Obstacle to Machine Translation*, "Advances in Computers", 1, rist. in Bar-Hillel(ed.)(1964), pp. 75-86.
- BAR-HILLEL Y.(ed.)(1964), *Language and Information*, Addison-Wesley, Palo Alto(Mass.).
- BAR-HILLEL Y., GAIFMAN C., SHAMIR E. (1960), *On Categorical and Phrase Structure Grammars*, "The Bulletin of the Research Council of Israel", pp. 1-16, rist. in Bar-Hillel(ed.)(1964), pp. 99-115.
- BERNERS-LEE T. (1994), *Issues for the future of the Web*, Opening keynote at the First International World Wide Web Conference, May 1994, Geneva, <http://www.w3.org/2001/sw/Europe/graphics/tmbtext.html>.
- BERNERS-LEE T., HENDLER J., ET AL. (2001), *The semantic web*, "Scientific American" 284 (5), pp. 34-43.
- BARSALOU L. W. (1987), *Ideals, Central Tendency, and Frequency of Instantiation as Determinants of Graded Structure in Categories*, "Journal of Experimental Psychology: Learning, Memory, and Cognition", 11(4), pp. 629-654.
- BARSALOU L. W., SEWELL D. R. (1985), *Contrasting the Representation of Scripts and Categories*, "Journal of Memory and Language", 24(6), pp. 646-665.

- BAZZANELLA C. (2005), *Linguistica e pragmatica del linguaggio*, Editori Laterza, Roma.
- BECHTEL W., GRAHAM G. (1998), *A Companion to Cognitive Science*, Blackwell, Oxford
- BETH E., PIAGET J. (1966), *Mathematical Epistemology and Psychology*, Reidel Dordrecht, Holland.
- BONIOLO G., VIDALI P. (2002), *Strumenti per ragionare*, Bruno Mondadori, Milano.
- BÖHM C., G. JACOPINI (1966), *Flow diagrams, Turing Machines and Languages with only Two Formation Rules*, Comm. of the ACM, 9(5), pp. 366-371.
- BRUNER J. S., GOODNOW J. J., AUSTIN G. A., BROWN R. W. (1956), *A Study of Thinking*, Wiley, New York; trad. it. *Il pensiero: strategie e categorie*, Armando, Roma, 1969.
- BUZZONI M. (2004), *Esperimento ed esperimento mentale*, Franco Angeli, Milano.
- CASADIO C. (1987), *Semantic Categories and the Development of Categorical Grammars*, in R. T. Oehrle, E. Bach and D. Wheeler (eds.)(1987), *Categorical Grammars and Natural Languages Structures*, Reidel, Dordrecht.
- ID. (1995), *Carnap e le lingue naturali*, in A. Pasquinelli (a cura di)(1995), *L'eredità di Rudolf Carnap*, CLUEB, Bologna, pp. 413-437.
- ID. (2005), *Logica e psicologia del pensiero*, Carocci, Roma.
- CARNAP R. (1934), *Logische Syntax der Sprache*, Vienna; trad. ing., *The Logical Syntax of Language*, London-New York, 1937; trad. it. di A. Pasquinelli, *La sintassi logica del linguaggio*, Silva, Milano, 1966.
- CARNAP R. (1947), *Meaning and Necessity*, University of Chicago Press, Chicago, trad. it. *Significato e necessità*, La Nuova Italia, Firenze, 1976.
- CHERUBINI P. (2005), *Psicologia del pensiero*, Raffaello Cortina Editore, Milano.

- CHIERCHIA G. (1997), *Semantica*, Il Mulino, Bologna.
- CHOMSKY N. (1956), *Three Models for the description of Language*, Symposium on Information Theory, MIT Cambridge, Mass.; in Luce R. D., Bush R. R., Galanter E. (eds.)(1965), *Readings in Mathematical Psychology*, 11, New York, Wiley, 1965, pp. 105-124; trad. it., *Tre modelli per la descrizione della lingua*, in A. De Palma (a cura di)(1974), pp. 203-35.
- ID. (1957), *Syntactic Structures*, The Hague, Mouton; trad. it., *Le strutture della sintassi*, Laterza, Bari, 1970.
- ID. (1963), *On Certain Formal Properties of Grammars*, "Information and Control" 2, pp. 137-67; rist. in Luce R. D., Bush R. R., Galanter E. (eds.)(1963), *Handbook of Mathematical Psychology*, New York, J. Wiley and Sons; trad. it. in A. De Palma (a cura di)(1974), pp. 271-310.
- ID. (1965), *Aspects of the Theory of Syntax*, The MIT Press, Cambridge (Mass.).
- ID. (1968), *Language and Mind*, Harcourt, Brace & World, New York.
- ID. (1977), *The Logical Structure of Linguistic Theory*, Plenum Press, London-New York.
- ID. (1981), *Lectures on Government and Binding*, Foris P.C., Dordrecht.
- ID. (2010), *Il linguaggio e la mente*, trad. it. di A. de Palma, Bollati Boringhieri, Torino.
- CHURCH A. (1936a), *An unsolvable problem of elementary number theory*, "American Journal of Mathematics" 58, pp. 345-364; ristampato in Davis M. (1965)(ed.), *The Undecidable*, Raven Press, New York, pp. 89-107.
- ID. (1936b), *A note of the Entscheidungsproblem*, "Journal of Symbolic Logic" 1, pp. 40-41; ristampato in Davis M. (1965)(ed.), *The Undecidable*, Raven Press, New York, pp. 110-115.
- COPI M., COHEN C. (1998), *Introduction to Logic*, Prentice-Hall, trad. it. *Introduzione alla logica*, Il Mulino, Bologna, 1999.
- COPELAND B. J. (1993), *Artificial Intelligence. A Philosophical Introduction*, Blackwell, Oxford.

- CRAICK K. J. W. (1943), *The Nature of Explanation*, Cambridge University Press; trad. it. parziale in Somenzi V. (a cura di), *La fisica della mente*, Boringhieri, Torino, 1969
- ID. (1947), *Theory of the Human Operator in Control Systems. Man as an Example in a Control System*, "British Journal of Psychology", 39, pp. 54-61,
- ID. (1966), *The Nature of Psychology*, Cambridge University Press; trad. it. cap. introd., *I meccanismi dell'azione umana*, in Somenzi e Cordeschi (1986), pp.63-85.
- CROFT W., CRUSE D. A. (2010), *Linguistica cognitiva*, Carocci Editore, Roma.
- CUMMINS R. (1989), *Meaning and Mental Representation*, The MIT Press, Cambridge (Mass.), trad. it. *Significato e rappresentazione mentale*, Il Mulino, Bologna, 1993.
- DALLA CHIARA C. (1967), *Contributo alla studio dei procedimenti di pensiero 'concreto' nell'adulto*, in "Rivista di Psicologia", 61, pp. 903-931.
- DALLA CHIARA M. L., GIUNTINI R., PAOLI F.(2004), *Sperimentare la logica*, Liguori Editore, Napoli.
- DAVIDSON D. (1980), *Essays on Actions and Events*, Clarendon Press, Oxford; trad. it. *Azioni ed eventi*, Il Mulino, Bologna.
- ID. (1984), *Inquiries into Truth and Interpretation*, Clarendon Press, Oxford; trad. it., *Verità e interpretazione*, Il Mulino, Bologna.
- DESCARTES R., *Opere 1637-1649*, Bompiani, Milano, 2009, pp. 2531; edizione integrale con traduzione italiana a fronte, a cura di G. Belgioioso.
- ID. *Opere 1637-1649*, Milano, Bompiani, 2009, pp. 2531; edizione integrale con traduzione italiana a fronte, a cura di G. Belgioioso
- ID. *Discorso sul metodo e altri scritti*, Bompiani, Milano, 2009.
- DE PALMA A. (a cura di)(1974), *La struttura logica del linguaggio*, Einaudi, Torino.

- DE VINCENZI M. (1989), *Syntactic Parsing Strategies in a Null Subject Language*, Doctoral Dissertation, University of Massachusetts, LSA Pub., Amherst (Mass.).
- ID. (1991), *Syntactic Parsing Strategies in Italian*, Kluwer Academic Pub., Dordrecht, the Netherlands.
- DE VINCENZI M., LOMBARDO V. (2000) *Cross-linguistic Perspectives on Language Processing*, Kluwer Academic Pub. Dordrecht, the Netherlands.
- EVANS J. ST. B. T. (1982), *The psychology of deductive reasoning*, Routledge & Kegan Paul, London.
- ID. (1983), *On the conflict between logic and belief in syllogistic reasoning*, in "Memory and Cognition", 11, pp. 295-306.
- ID. (1989), *Bias in Human Reasoning. Causes and Consequences*, Lawrence Erlbaum Associates, Hove, UK.
- ID. (2002), *Logic and human reasoning: an assessment of the deduction paradigm*, in "Psychological Bulletin", 128, pp. 978-996.
- ID. (2003), *In two minds: dual-process accounts of reasoning*, in "Trends in Cognitive Science", 7(10), 454-459.
- FODOR J. A. (1983), *The Modularity of Mind. An Essay on Faculty Psychology*, The MIT Press, Cambridge MA, trad. it. , *La modularità della mente*, Il Mulino, Bologna, 1988.
- ID. (2001), *The Mind Doesn't Work That Way*, The MIT Press, Cambridge, MA.
- ID. (2001), *Mente e linguaggio*, Editori Laterza, Roma.
- FODOR J. A., PYLYSHYN Z. W. (1988), *Connectionism and cognitive architecture: A critical analysis*, in "Cognition", 28, pp. 3-71.
- FREGE G. (1879), *Begriffsschrift*, in I. Angelelli (ed.), *Wissenschaft. Buchgesellschaft*, Darmstadt, 1973; trad. it. in C. Mangione (a cura di)(1965), *Logica e aritmetica*, Boringhieri, Torino.

- ID. (1892), *Über Sinn und Bedeutung*, in “Zeitschrift für Philosophie und philosophische Kritik”, 100, pp.25-50; trad. it. *Senso e denotazione*, in A. Bonomi (a cura di)(1973), *La struttura logica del linguaggio*, Bompiani, Milano, pp. 9-32.
- FRIXIONE M., PALLADINO D. (2004), *Funzioni, Macchine, Algoritmi. Introduzione alla teoria della computabilità*, Carocci, Roma.
- GARNHAM A., OAKHILL J. V. (1994), *Thinking and Reasoning*, Basil Blackwell, Oxford.
- GAZZANIGA M. S. (1985), *The Social Brain*, Basic Books, New York.
- GILINSKY A. S., JUDD B. B. (1994), *Working memory and bias in reasoning across the life-span*, in “Psychology of Aging”, 9, pp. 356-371.
- GILOVICH T., GRIFFIN D. (2002), *Introduction - Heuristics and biases: then and now*, in T. Gilovich et al. (eds.), *Heuristics and Biases: The Psychology of Intuitive Judgement*, pp. 1-18, Cambridge University Press, Cambridge.
- GIROTTO V. (1994), *Il ragionamento*, Il Mulino, Bologna.
- GIROTTO V., GONZALEZ M. (1999), *Strategies and models in statistical reasoning*, in W. Schaeken, G. De Vooght, A. Vandierendonck, G. D’Ydewalle (eds.), *Strategies in Deductive Reasoning*, Lawrence Erlbaum Associates, Mahwah, NJ.
- GIROTTO V., KEMMELMEIER M., SPERBER D., VAN DER HENST J.-B. (2001), *Inept reasoners or pragmatic virtuosos? Relevance and the deontic selection task*, in “Cognition” 81, pp. B69-B76.
- GIROTTO V., LEGRENZI P. (1999)(a cura di), *Psicologia del pensiero*, Il Mulino, Bologna.
- GIROTTO V., POLITZER G. (1990), *Conversational and world-knowledge constraints in deductive reasoning*, in J.P. Caverni, J.M. Fabre, M. Gonzalez (eds.), *Cognitive Biases*, North-Holland, Amsterdam.
- GIROTTO V., LEGRENZI P. (2000), *Psicologia del pensiero*, Il Mulino, Bologna.

- GLYMOUR C. (1992), *Thinking Things Through*, MIT, Cambridge (Mass.), trad. it. *Dimostrare, credere, pensare. Introduzione all'epistemologia*, R. Cortina Editore, Milano, 1999.
- GOEL V. (1995), *Sketches of Thought*, MIT Press, Cambridge, MA.
- GOEL V., BUCHEL C., FRITH C., DOLAN R. J. (2000), *Dissociation of Mechanisms Underlying Syllogistic Reasoning* in "NeuroImage" 12, pp. 504-514.
- GOEL V., DOLAN R. J. (2001), *Functional Neuroanatomy of Three-term Relational Reasoning*, in "Neuropsychologia" 39, pp. 901-909.
- GOEL V., DOLAN R. J. (2004), *Differential involvement of left prefrontal cortex in inductive and deductive reasoning*, in "Cognition", 93, B109-B121.
- GOEL V., GOLD, B., KAPUR S., HOULE S. (1998), *Neuroanatomical Correlates of Human Reasoning*, in "Journal of Cognitive Neuroscience", 10(3), pp. 293-3.
- GRICE P. (1975), *Logic and Conversation*, in P. Cole, J. Morgan (eds.), *Syntax and Semantics 3: Speech Acts*, Academic Press, New York; trad. it., *Logica e conversazione*, in M. Sbisà (a cura di), *Gli atti linguistici*, Feltrinelli, Milano, 1978.
- HAREL D, FELDMAN Y. (2008), *Algoritmi. Lo spirito dell'informatica*, Springer-Verlag, Milano; tit. orig. *Algorithmics. The spirit of computing*, Addison Wesley, 1992.
- HODGES A. (2007), *Alan Turing*, The Stanford Encyclopedia of Philosophy, E. N. Zalta (ed.), URL = <<http://plato.stanford.edu/archives/sum2011/entries/turing/>>.
- HOPCROFT J. E., ULLMAN J. D. (1979), *Formal Languages and their Relation to Automata*, Addison-Wesley, Reading, MA.
- HULL C. L. (1930), *Knowledge and Purpose as Habit Mechanism*, "Psychological Review", 37 (6), pp. 511-23.
- ID. (1943), *Principles of Behavior*, Appleton Century, New York; tad. It. *I principi del comportamento*, Armando, Roma.

- HUNT E. B. (1999), *What is a theory of thought?*, in R. J. Sternberg (ed.), *The nature of cognition*, pp. 3-49, The MIT Press, Cambridge MA.
- JOHNSON-LAIRD P. N. (1983), *Mental models*, Cambridge University Press, Cambridge, Harvard University Press, Cambridge MA.
- JOHNSON-LAIRD P. N. (1991), *Human and machine thinking*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- ID. (2001), *Mental models and deductive reasoning*, in "Trends in Cognitive Science", 5, pp. 434-442.
- JOHNSON-LAIRD P. N., LEGRENZI P., GIROTTO V., LEGRENZI M. S. (2000), *Illusions in reasoning about consistency*, in "Science", 288, pp. 531-532.
- JOHNSON-LAIRD P. N., LEGRENZI P., GIROTTO V. (2004), *Reasoning from inconsistency to consistency*, in "Psychological Review", 111, n. 3.
- KAHNEMAN D. (2000), *A psychological point of view: Violations of rational rules as a diagnostic of mental processes (Commentary on Stanovich and West)*, in "Behavioral and Brain Sciences", 23, pp. 681-683.
- KAHNEMAN D., FREDERICK S. (2002), *Representativeness revisited: attribute substitution in intuitive judgment*, T. Gilovich, D. Griffin and D. Kahneman (eds.), *Heuristics of Intuitive Judgment: Extensions and Applications*, pp. 49-81, Cambridge University Press, New York.
- KAHNEMAN D., TVERSKY A. (1996), *On the reality of cognitive illusions: A reply to Gigerenzer's critique*, in "Psychological Review", 103, pp. 582-591.
- KAHNEMAN D., SLOVIC P., TVERSKY A. (1982), *Judgment Under Uncertainty: Heuristics and Biases*, Cambridge University Press, New York.
- KNEALE W. C., KNEALE M. (1972), *Storia della logica*, Einaudi, Torino.
- KUHN D., KATZ J. B., DEAN D. JR. (2004), *Developing reason*, in "Thinking and Reasoning", 10(2), pp. 197-219.
- LA METTRIE J. O. (1745), *Histoire naturelle de l'âme*, Paris; vedi *Opere Filosofiche*, Laterza, Bari, 1992.

- LAMBEK J. (1958), *The Mathematics of Sentence Structure*, "American Math. Monthly", 65, pp.154-170; trad. it. *La matematica della struttura di frase*, in A. De Palma (a cura di)(1974), pp. 241-264.
- ID. (1961), *On the Calculus of Syntactic Types*, in R. Jacobson (ed.)(1961), *Structure of Language and its Mathematical Aspects*, Providence, pp.166-178.
- LEGRENZI, P. (1971), *Discovery as a means to understanding*, in "Quarterly Journal of Experimental Psychology", 23, pp. 417-422.
- ID. (2001), *Psicologia cognitiva applicata*, Editori Laterza, Roma.
- ID. (2002), *Prima lezione di scienze cognitive*, Editori Laterza, Roma.
- LEGRENZI P., GIROTTO V., JOHNSON-LAIRD P. N. (1993), *Focussing in reasoning and decision making*, in "Cognition", 49 , pp. 37-66.
- ID. (2003), *Models of Consistency*, "Psychological Science", 14(2), 131-137.
- MCCULLOCH W. S., PITTS W. H. (1943), *A logical calculus of the ideas immanent in nervous activity*, "Bulletin of Mathematical Biophysics", 5, pp.115-133.
- MACCHI L. (1999), *Formazione dei concetti e processi di categorizzazione*, in Girotto V. e Legrenzi P. (a cura di), *Psicologia del pensiero*, Il Mulino, Bologna, pp. 91-116.
- MANGIONE C., BOZZI S. (1993), *Storia della logica da Boole ai giorni nostri*, Garzanti, Milano.
- MANKTELOW K. (1999), *Reasoning and Thinking*, Cognitive Psychology, A Modular Course, Psychology Press, Hove UK.
- MARKOVITS H., BARROUILLET P. (2004), *Introduction: Why is understanding the development of reasoning important?*, in "Thinking and Reasoning", 10(2), pp. 113-121.
- MAREK P., GRIGGS R. A., KOENIG C. S. (2000), *Reducing cognitive complexity in a hypothetico-deductive reasoning task*, in "Thinking and Reasoning", 6 (3), pp. 253-265.

- MARLETTI C., MORICONI E., MARIANI M. (2009), *Argomenti di logica*, Edizioni PLUS, Pisa University Press, Pisa.
- MARRAFFA M. (2002), *Scienza cognitiva. Un'introduzione filosofica*, CLEUP, Padova.
- ID. (2003), *Filosofia della psicologia*, Editori Laterza, Roma.
- MARRAFFA M., MEINI C. (2005), *La mente sociale. Le basi cognitive della comunicazione*, Editori Laterza, Roma.
- MENDELSON E. (1972), *Introduzione alla logica matematica*, Boringhieri, Torino.
- MILLER G. A. (1956), *The magical number seven, plus or minus two: some limits on our capacity for processing information*, "Psychological review", 63, pp. 81-97.
- MINSKY, M. L. (1959), *Some Methods of Artificial Intelligence and Heuristic Programming*, in D. V. Blacke & A. M. Uttley (eds.), *Proceedings of the Symposium on Mechanisation of Thought Processes*; trad. it. *Intelligenza artificiale e programmazione euristica*, in Somenzi e Cordeschi (1986), pp. 194-203.
- ID. (1968)(ed.), *Semantic information processing*, Cambridge, MA, MIT Press.
- ID. (1985), *The Society of Mind*, Simon & Schuster, New York; trad. it. *La società della mente*, Adelphi, Milano, 1989.
- MONTAGUE R. (1974), *Formal Philosophy*, New Haven, Yale University Press.
- MORICONI E. (2009), *Strutture dell'argomentare*, Edizioni ETS, Pisa.
- NEUMANN VON J. (1958), *The Computer and the Brain*, Yale University Press; trad. it, *Il calcolatore e il cervello*, in Somenzi e Cordeschi (1986), pp. 108-150).
- NEWSTEAD S. E., ELLIS M. C., EVANS J. ST. B. T., DENNIS I. (1997), *Conditional reasoning with realistic material*, in "Thinking and Reasoning", 3(1), pp. 49-76.

- NEWELL A. (1981), *Reasoning, problem solving and decision processes: the problem space as a fundamental category*, in R. Nickerson (ed.), *Attention and Performance*, Vol. 8, Lawrence Erlbaum Associates, Hillsdale, NJ.
- ID. (1990), *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.
- NEWELL A., SIMON H. A. (1972), *Human Problem Solving*, Englewood Cliffs, Prentice-Hall.
- NICKERSON R. S. (1994), *The teaching of thinking and problem solving*, in R. J. Sternberg (ed.), *Thinking and problem solving*, pp. 409-449, Academic Press, San Diego.
- ID. (1996), *Hempel's Paradox and Wason's Selection Task: Logical and Psychological Puzzles of Confirmation*, in "Thinking and Reasoning", 2(1), pp. 1-31.
- NUMERICO T. (2005), *Alan Turing e l'intelligenza delle macchine*, Franco Angeli, Milano
- OAKSFORD M., CHATER N. (1994), *A rational analysis of the selection task as optimal data selection*, in "Psychological Review", 101, pp. 608-631.
- OAKSFORD M., STENNING K. (1992), *Reasoning with conditional containing negated constituents*, in "Journal of Experimental Psychology: Learning, Memory and Cognition", 18, pp. 835-854.
- OPPY G., DOWE D. (2011), *The Turing Test*, The Stanford Encyclopedia of Philosophy, E. N. Zalta (ed.), URL: <<http://plato.stanford.edu/archives/spr2011/>>.
- PALLADINO D., (2002), *Corso di logica. Introduzione elementare al calcolo dei predicati*, Carocci Editore, Roma.
- ID., (2004), *Logica e teorie formalizzate. Completezza, incompletezza, indecidibilità*, Carocci Editore, Roma.
- PESSA E., PENNA M. P. (2000), *Manuale di scienza cognitiva. Intelligenza artificiale classica e psicologia cognitiva*, Editori Laterza, Roma.

- PIAGET J. (1947), *La psychologie de l'intelligence*, Armand Colin Publisher, 1998; trad. it. *Psicologia dell'intelligenza*, Giunti, Firenze, 2011.
- PICARDI E. (1992), *Linguaggio e analisi filosofica*, Patron, Bologna.
- ID. (1999), *Le teorie del significato*, Editori Laterza, Roma.
- ID. (2001), *Frege. Senso, funzione e concetto*, Editori Laterza, Roma-Bari.
- PINKER S. (1994), *The Language Instinct: How the Mind Creates the Language*, Penguin Books, London; trad. it. *L'istinto del linguaggio. Come la mente crea il linguaggio*, Mondadori, Milano, 1997.
- PLATONE, *Opere complete*, vol. II. *Cratilo, Teeteto, Sofista, Politico*; trad. di L. Minio-Paluello, M. Valgimigli, A. Zadro, Biblioteca Universale Laterza, 2003.
- POPPER K. R. & ECCLES J. C. (1977), *The Self and its Brain. An Argument for Interactionism*, Springer-Verlag, London & Berlin; trad. it. *L'io e il suo cervello*; vol. 1, *Materia, coscienza e cultura*; vol. 2, *Strutture e funzioni cerebrali*, Armando Editore, Roma, 1981 (5^a rist. 2001).
- REALE G., ANTISERI D. (2009), *Storia della Filosofia* vol. 1 - *Dai Presocratici ad Aristotele*, Bompiani, Milano:
- RIPS L. J. (1994), *The Psychology of Proof*, MIT Press, Cambridge, MA.
- ROMANO M. (2011), *Ontologies, Logic and Interaction*, PhD Thesis, Univ. Roma Tre.
- ROOKES P., WILLSON J. (2000), *Perception*, Routledge, London; trad. it. *La percezione*, Il Mulino, Bologna.
- ROSCH E. (1977), *Human categorization*, in Warren N. (ed.), *Studies in Cross Cultural Psychology*, Academic prss, London, vol 1.
- ID. (1978), *Principles of Categorization*, in E. Rosh, B. B. Loyd (eds.), *Cognition and Categorization*, Erlbaum, Hillsdale N. J.
- ID. (1983), *Prototype Classification and Logical Classification: The Two Systems*, in E. K. Scholnick (ed.), *New Trends in Conceptual Representation: Challenges to Piaget's Theory?*, Erlbaum, Hillsdale, N. J., pp. 73-86.

- ROSCH E., MERVIS C. B. (1975), *Family resemblance: Studies in the internal structure of categories*, "Cognitive Psychology", 7, 573-605.
- ROSENBLUETH A., WIENER N., BIGELOW J. (1943), *Behavior, Purpose and Teleology*, "Philosophy of Science", 10 (1), pp. 18-24; trad. it. In Somenzi e Cordeschi (1986), *Comportamento, scopo e teleologia*, pp. 78-85.
- ROSS B. H., SPALDING T. L. (1994), *Concepts and categories*, in R. J. Sternberg (ed.), *Thinking and problem solving*, New York, Academic Press, pp. 119 - 148.
- RUSSELL B. (1918), *The Philosophy of Logical Atomism*, in *The Collected Papers of Bertrand Russell, 1914-19*, vol.8, London, Allen & Unwin.
- SEARLE J. L. (1980), *Minds, brains, and programs*, "Behavioral and Brain Sciences", 3 (3), pp. 417-457.
- ID. (1998), *Mind, Language and Society*, trad. it. *Mente, linguaggio, società. La filosofia nel mondo reale*, R. Cortina Editore, Milano, 2000.
- SHANNON C. E. (1938), *A Symbolic Analysis of Relay and Switching Circuits*, "Transactions American Institute of Electrical Engineers", vol. 57, pp. 571-495; abstract from the thesis presented in 1936 at MIT, Mass.
- ID. (1948), *A Mathematical Theory of Communication*, Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656.
- ID. (1953), *Computers and Automata*, Proc. IRE, vol. 41, pp. 1234-41; trad. it. *Calcolatori e automi*, in Somenzi e Cordeschi (1986), pp. 93-107.
- SHANNON C. E., WEAVER W. (1949), *The Mathematical Theory of Communication*, The University of Illinois Press, Urbana, Illinois.
- SOMENZI V., CORDESCHI R. (1986), *La filosofia degli automi. Origini dell'intelligenza artificiale*, Bollati Boringhieri, Torino.
- SOWA J. F. (1984), *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA.
- SPERBER D., WILSON D. (1986), *Relevance*, Basil Blackwell, Oxford.

- STANOVICH K. E., WEST R. F. (2001), *Individual differences in reasoning: Implications for the rationality debate?*, in "Behavioral and Brain Sciences", 23, pp. 645-726.
- STENNING K., VAN LAMBALGEN M. (2001), *Semantics as a foundation for Psychology: a case study of Wason's selection task*, in "Journal of Logic, Language and Information", 10, pp. 273-317.
- ID. (2004), *A little logic goes a long way: basing experiment on semantic theory in the cognitive science of conditional reasoning*, in "Cognitive Science", 28, pp. 481-529.
- STERNBERG R. J. (1977), *Intelligence, information-processing, and analogical reasoning: The componential analysis of human abilities*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- ID. (1983), *Components of human intelligence*, in "Cognition", 15, pp. 1-48.
- ID. (ed.) (1984), *Human Abilities: An Information-processing Approach*. Freeman, San Francisco.
- TURING A. (1937), *On computable numbers, with an application to the Entscheidungsproblem*, "Proceedings of the London Mathematical Society", ser. 2, vol. 42, 1936-37, pp. 230-265; ristampato in Davis M. (1965)(ed.), *The Undecidable*, Raven Press, New York, pp. 116-154.
- ID. (1950), *Computing machinery and intelligence*, "Mind" 59, pp. 433-60; trad. it. *Macchine calcolatrici e intelligenza*, in Somenzi e Cordeschi (1986), pp. 167-193.
- TVERSKY A., KAHNEMAN D. (1974), *Judgment under uncertainty: heuristics and biases*, in "Science", 185, pp. 1124-1131.
- WASON P. C. (1968), *Reasoning about a rule*, in "Quarterly Journal of Experimental Psychology", 20, pp. 273-281.
- ID. (1969), *Regression in reasoning*, in "British Journal of Psychology", 60, pp. 471-480.

- WASON P. C., JONES S. (1963), *Negatives: denotation and connotation*, in "British Journal of Psychology", 54(4), pp. 299-307.
- WASON P. C., KOSVINER A. (1966), *Perceptual distortion induced by reasoning*, in "British Journal of Psychology", 57(3), pp. 413-418.
- WASON P. C., JOHNSON-LAIRD P. N. (eds.)(1968), *Thinking and Reasoning*, Penguin Books Ltd, Harmondsworth, UK.
- ID. (1970), *A conflict between selecting and evaluating information in an inferential task*, in "British Journal of Psychology", 61, pp. 509-515.
- ID. (1972), *The psychology of deduction: Structure and content*, Harvard University Press, Cambridge, MA.
- WHITEHEAD A. N., RUSSELL B. (1910/1913), *Principia Mathematica*, II ed. 1924, Cambridge.
- WIENER N. (1948), *Cybernetics, or Control and Communications in the Animal and the Machine*, The MIT Press, Cambridge, Mass. (2nd ed. 1961); trad. it. *Cibernetica*, Il Saggiatore, Milano, 1968.
- ID. (1950), *The Human Use of Human Beings*, Boston; trad. it. *Introduzione alla cibernetica. L'uso umano degli esseri umani*, Universale Scientifica Boringhieri, Torino, 1966.
- WITTGENSTEIN L. (1953), *Philosophische Untersuchungen*, Blackwell, Oxford; trad. it. *Ricerche filosofiche*, Einaudi, Torino, 1995.